2001-09

# Hi-fidelity simulation and prediction of helicopter single point external load stabilization

Ehlers, George E.

Monterey, California. Naval Postgraduate School

# NAVAL POSTGRADUATE SCHOOL
## Monterey, California

# THESIS

**HI-FIDELITY SIMULATION AND PREDICTION OF HELICOPTER SINGLE POINT EXTERNAL LOAD STABILIZATION**

by

George E. Ehlers

September 2001

Thesis Co-Advisors: E. Roberts Wood
Mark B. Tischler

| REPORT DOCUMENTATION PAGE | *Form Approved OMB No. 0704-0188* |
|---|---|

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

| **1. AGENCY USE ONLY *(Leave blank)*** | **2. REPORT DATE** September 2001 | **3. REPORT TYPE AND DATES COVERED** Master's Thesis |
|---|---|---|

| | |
|---|---|
| **4. TITLE AND SUBTITLE**: Title (Mix case letters) Hi-Fidelity Simulation and Prediction of Helicopter Single Point Eternal Load Stabilization. | **5. FUNDING NUMBERS** |
| **6. AUTHOR(S)** Ehlers, George E | |
| **7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)** Naval Postgraduate School Monterey, CA 93943-5000 | **8. PERFORMING ORGANIZATION REPORT NUMBER** |
| **9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)** N/A | **10. SPONSORING / MONITORING AGENCY REPORT NUMBER** |

**11. SUPPLEMENTARY NOTES** The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

| **12a. DISTRIBUTION / AVAILABILITY STATEMENT** Approved for public release; distribution is unlimited | **12b. DISTRIBUTION CODE** |
|---|---|


**13. ABSTRACT *(maximum 200 words)***

The helicopter has been used since its early development for external transport of large or bulky loads to small austere locations. Among the problems encountered as lift capability and airspeeds increased was that of divergent load oscillations due to load aerodynamics

The most problematic are the single point external loads displaying unsteady aerodynamics and coupled yaw-pendulum modes accounting for the instability of cargo containers. However, a lack of simulation models for unsteady aerodynamics renders simulation and analysis incapable of predicting the critical speeds at which such loads become unstable. This thesis attempts to provide a stabilization system for controlling the yaw degree of freedom for the single point external load.

Empirical models of the yaw resistance at the hook and of the yaw moments due to vortex shedding were developed and tuned using flight test data and lab measurements. Several load stabilization systems were considered, and a horizontal and vertical tail fin assembly was selected This thesis presents simulation model improvements required for a simulation to match flight results for the load yaw, along with the design, modeling and optimization of the fin stabilization system, and a simulation assessment of the envelope expansion obtained from both passive and active stabilization.


| **14. SUBJECT TERMS** Air Vehicles, Helicopter/Slung Load, External Load, Flight Simulation | **15. NUMBER OF PAGES** 187 |
|---|---|
| | **16. PRICE CODE** |

| **17. SECURITY CLASSIFICATION OF REPORT** | **18. SECURITY CLASSIFICATION OF THIS PAGE** | **19. SECURITY CLASSIFICATION OF ABSTRACT** | **20. LIMITATION OF ABSTRACT** |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | UL |

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release; distribution is unlimited**

# HI-FIDELITY SIMULATION AND PREDICTION OF HELICOPTER SINGLE POINT EXTERNAL LOAD STABILIZATION

George E. Ehlers
Major, United States Marine Corps
Batchelor of Science, University of Nebraska-Lincoln, 1991

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING**

from the

**NAVAL POSTGRADUATE SCHOOL**
**September 2001**

Author: George E Ehlers

Approved by: E. Roberts Wood, Thesis Co-Advisor

Mark B. Tischler, Thesis Co-Advisor

Max F. Platzer, Chairman
Department of Aeronautics and Astronautics

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

The helicopter has been used since its early development for external transport of large or bulky loads to small austere locations. Among the problems encountered as lift capability and airspeeds increased was that of divergent load oscillations due to load aerodynamics

The most problematic are the single point external loads displaying unsteady aerodynamics and coupled yaw-pendulum modes accounting for the instability of cargo containers. However, a lack of simulation models for unsteady aerodynamics renders simulation and analysis incapable of predicting the critical speeds at which such loads become unstable. This thesis attempts to provide a stabilization system for controlling the yaw degree of freedom for the single point external load.

Empirical models of the yaw resistance at the hook and of the yaw moments due to vortex shedding were developed and tuned using flight test data and lab measurements. Several load stabilization systems were considered, and a horizontal and vertical tail fin assembly was selected This thesis presents simulation model improvements required for a simulation to match flight results for the load yaw, along with the design, modeling and optimization of the fin stabilization system, and a simulation assessment of the envelope expansion obtained from both passive and active stabilization.

THIS PAGE INTENTIONALLY LEFT BLANK

# TABLE OF CONTENTS

# LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

# ACKNOWLEDGMENTS

The author would like to acknowledge the financial support of the Army/NASA Rotorcraft Division, for providing the necessary travel and lodging funds for an internship at the Ames Research Center, and many trips between Monterey and San Jose.

Special thanks go to my thesis advisors, Dr. E. Roberts Wood and Dr. Mark Tischler for their expert guidance and insights. Also to Mr Luigi Cicolani for his untiring guidance in the conduct of this research and outstanding assistance in the editing of this thesis.

Lastly, I thank my wife and daughters, who allowed me the time to conduct the internship and whose support was essential to the research's successful conclusion.

THIS PAGE INTENTIONALLY LEFT BLANK

# I. INTRODUCTION

## A. HISTORICAL

The helicopter has been used since its early development for external transport of large or bulky loads to small austere locations. For the military it has been essential in rapidly providing supplies not only for combat but also for natural disasters where landing locations are small and isolated. In addition the ability of the helicopter to transport loads externally has changed the face of construction, logging and fire fighting by providing a mobile heavy lift capability.

As helicopters improved, their lift capability and airspeeds increased to encounter the problem of load aerodynamic instability. According to various published works [Ref 1-4] aerodynamic instabilities can occur at moderate forward airspeeds well below the power-limited airspeed of today's helicopters.

In order to fly a helicopter safely with a slung load (Helo/SL) the handling qualities and maximum forward speed of the combination has to be determined. This has normally been done by flight-testing of the Helo/SL and has led to millions of dollars being spent by the military and industry to certify new loads and cargo hooks.

Wind tunnel studies [Ref 5] shows that many loads become unstable at airspeeds well below a helicopter's power limited speed. This restriction severely hampers a helicopter's ability to rapidly re-supply time critical situations such as humanitarian relief or fire fighting where needed supplies may be a considerable distance away.

## B. LOAD STABILIZATION

The simplest, most popular and problematic external load suspension system is the single point configuration. From the early use it became apparent that a method of stabilizing this configuration to allow the helicopter to fly at higher airspeeds would be a great improvement to the Helo/SL. With no inherent yaw stabilization the single point external load it displays potentially dangerous yaw to lateral coupling which leads to pendulous oscillations, load helicopter contact, load jettison and even loss of the helicopter or life can be controlled, greatly improving the safety and speed of the Helo/SL operations.

In addition an accurate, validated simulation would allow various loads to be checked with a variety of sling configurations or even with load stabilization devices to determine the optimum configuration. This would allow for a reduction in the flight test time required to certify loads and would improve safety and training for external load operations.

## C. PREVIOUS RESEARCH

Historically slung load stabilization has been approached in three basic methods; altering the original control systems of the helicopter, altering the geometry and by attaching various stabilization devices directly to the load.

Altering the original control systems of the helicopter such as altering the Augmented Flight Control System control laws to include feedback of yaw motions and have shown limited success [Ref 6] but adds an additional level of complication and concerns for flight safety.

Altering the sling configuration such as the flight test of the active arm stabilization system developed by Boeing [Ref 7] has shown promising results but adds considerable bulk and weight when added to existing aircraft. This method appears to be most promising if it is incorporated during the design of the next generation of heavy lift helicopters.

Attaching various stabilization devices directly on the load such as the early experimentation with drogue shoots and flow diverters has been frequently proposed and studied and has shown promising results. This method of stabilization is a logical approach since these devices attack the source of the instability directly, namely the load aerodynamics and do not require modifications of flight control systems.

## D. STABILIZATION DEVICES

Proposed load stabilization devices have included a rotating wheel mounted on the load to augment stability [Ref 8] and active jet controls mounted on the helicopter's lower pennant hook [Ref 9] and load shape modifications [Ref 10]. While all of these methods show some improvement in theoretical analysis, from the practical standpoint the increased weight and complexity does not appear to make these methods practical at this time

Liu [Ref 10] examined the effects of changing load stability parameters, which can be done by adding fore and after body flow directors or by adding weight to the load. His analytical results showed a limited effect on load stability and since these types of stabilization devices are very load specific, their practicality is also limited. However Liu also showed that the most significant increase in stability was achieved by reducing the coefficient of yaw due to beta ($C_{\eta_\beta}$). This can be done by adding fins to the load. Further studies by Mtheson and by Gera and Farmer [Ref 11,12] showed that controllable fins could provide even greater load stability than static fins. This appears to be the most promising and practical approach and was chosen for this research.

## E. PURPOSE OF RESEARCH

Slung load research at NASA Ames research center in single point suspension systems have been conducted since 1995 under a Memorandum of Agreement between the United States and Israel [Ref 13]. This research on rotorcraft aeromechanics and man-machine integration covers many topics including the flight mechanics of helicopters / sling-load systems. Early objectives of this combined research have largely been accomplished and includ static wind tunnel tests of cubes and CONEX model [Ref 14], flight tests of various single point external loads [Ref 15] and a simulation of the helicopter / slung load two-body system [Ref 16]. Current objectives are to provide a validated helicopter / slung load simulation for existing configurations and to provide a dynamic prediction tool for new configurations.

The first goal served three additional functions. A validated simulation model could then be used to drive a motion-based flight simulator used for realistic pilot training, providing the correct "feel" of the external load operations. Secondly, the model could be used for load certification and configuration experimentation. Finally, provided valuable insight into conditions under which a particular load goes unstable. Understanding what operational techniques delay the onset of instability could expand the flight envelopes for existing loads. A larger flight envelope increases efficiency, important factor for civil and military slung load operations.

In the design stage of a new helicopter or a new load configuration, a dynamic prediction tool would aid in verifying safety of flight prior to expensive wind tunnel or flight-testing. A simulation with modular architecture, allowing the designer to assess different elements representing helicopter dynamics, load aerodynamics, and sling geometry, would make possible extensive parametric studies, showing the sensitivity of the complete design to variations in any of the basic elements of the system.

## F. SCOPE OF CURRENT THESIS WORK PERFORMED

This thesis describes the development and analysis of a fin type load stabilization device using a comprehensive dynamics and aerodynamics model for slung load simulation previously developed by the NASA Ames rotorcraft division [Ref 16]. This simulation is an integration of the NASA Ames GenHel UH-60A simulation with the equations of motion for slung load systems it includes in;

- Static aerodynamics review.
- Unsteady aerodynamics review.
- Yaw Motion Correction and validation to the GenHel/Slung Load simulation.
- Theoretical design and analysis of a fin type stabilization system.

Gen Hel is a component-type nonlinear Black Hawk helicopter model, using a blade-element implementation of the main rotor system. Devloped by Sikorsky [Ref 17] and validated at NASA Ames for handling qualities research it's main rotor dynamics in the model include rotor flapping, lagging, and rotor speed, incorporating lag dampers to enable the model to accurately predict possible rotor instability due to air resonance.

The slung load dynamics model was integrated into the GenHel/SL simulation by Tyson [Ref 17] from previous work at Ames [Ref 26]. This model used flight test data to validate the motions of the load. The results showed accurate reproduction of the load longitudinal and lateral pendulum motions [Ref 15, 16] but the yaw motions seen in flight were not matched.

An in-depth review of the static and dynamic aerodynamics of the slung load was conducted to gain an appreciation of the forces behind the load's motion. Basic airflow analysis along with flow separation and vortex shedding was considered to develop a mathematical model which closely followed both the steady aerodynamic wind tunnel data obtained at the Technion Institute, Israel [Ref 14] and flight test data obtained at the Ames Research Center [Ref 15].

The aerodynamics review was used to obtain an empirical formula to model the apparent aerodynamic forces on the load and the formula expanded to include the yaw resistance at the hook. Laboratory experiments were conducted to obtain the actual yaw resistance at the hook and the simulation validated against flight test data.

A load stabilization module was developed and linked to the GenHel/SL simulation. The stabilizer aerodynamics was modeled for an airfoil exposed to a relative wind from any angle. The affects of a stabilization device of various sizes were studied using the CONEX box as load.

Historically three basic types of loads have displayed a tendency towards instability; bluff bodies, flat plate type and aerodynamically active loads. Though this work focuses primarily on the bluff body loads the model was developed with the idea of expanding it to all types of unstable loads.

The static and unsteady aerodynamics of a bluff body are evaluated in chapter II in order to get an appreciation of the forces causing the yaw motions. Past stabilization studies were reviewed in chapter III and a stabilization device was selected based on operational considerations. The yaw motion of the GenHel/SL simulation was corrected and validated in chapter IV using flight test data. A stabilization module was developed in chapter V and linked to the GenHel/SL simulation and the effects of the stabilization system were evaluated.

# II. BLUFF BODY FORCES AND MOMENTS

## A. OVERVIEW

External load aerodynamics were considered prior to developing the load stabilization system model and a correction of its yaw motions. The aerodynamics accounts for the instability of many difficult loads at moderate forward speeds. The airflow around the load was considered in order to derive an appropriate simulation model

The basic nature of the load aerodynamics differs in hover and forward flight. During hover the airflow around the load is determined by the rotor downwash, the magnitude of which and the amount of swirl in the downwash being dependent on the thrust of the helicopter/slung load as well as the helicopter's rotor configuration. In forward flight the geometry and motions of the load determine the airflow around the load. For analysis the load aerodynamics were separated into steady and unsteady aerodynamics and the two effects are superposed.

The remaining force acting on the load is that of hook resistance. The single point suspension system can be connected to the helicopter directly or by using a swivel assembly each providing a unique resistance to the load motions. The resulting equation incorporated all aspects effecting the yaw motions of the load is;

$$YM = YM_{Static} + YM_{Dynamic} + YM_{Swirl} - YM_{Hook\,\mathrm{Re}sis\tan ce}$$

## B. HOVER AERODYNAMICS

The aerodynamics in a hover can be greatly simplified by only looking at those forces that cause motion in the external load. For this analysis the aircraft is considered to be in a hover when the aircraft is in a no-wind environment or that the wind is slight enough that it's affect on the load is negligible.

In evaluating effects of the downwash from the helicopter impinging on top of the load (uniform distribution) the only factor from the downwash that causes motion of the load is downwash swirl. In hover flight tests conducted at Ames with a swiveled sling, yaw rates of about 45 degrees per second were seen. The GenHel / slung load simulation from reference (16) modeled the rotor downwash axial flow and represented the swirl as a lateral air velocity component. Although the axial flow can be 50 knots, it had negligible effect on the pendulum roots. The simulation roots did match the flight data but could not accurately model the yaw motions.

## C. FLIGHT AERODYNAMICS

A bluff body aerodynamic analysis was performed for understanding of the nature of the box aerodynamics over a 360-degree range.

Most unstable bluff body loads are medium to low density box-type loads. These display evidence of close coupling between lateral pendulous motions and yaw [Ref 12]. For this analysis the aerodynamics of a typical box-type load was considered.

### 1. Wind Tunnel Data

An in-depth wind tunnel study of the static aerodynamics of cubic containers and the CONEX cargo container was conducted by Technion Israel [Ref 15]. This data was comprehensive and accurate. All angles-of-attack and sideslip, as required for a complete simulation model were covered. Departures of measurements from the expected symmetries of the cube were used to improve the model mounts and remove interference effects.

### 2. Steady Aerodynamic Analysis

A basic analysis of the steady airflow around a box of length l, width w, and height h, which is fixed in the airflow, is shown in figure (1). As the box turns in the airflow the streamlines will be deflected creating a lateral force perpendicular to the free stream velocity that will continue the motion until the two forces lie on a line through the geometric center. Any further rotation of the box beyond this angle of zero yaw moment $(\beta_0)$ will tend to return the load to this angle.

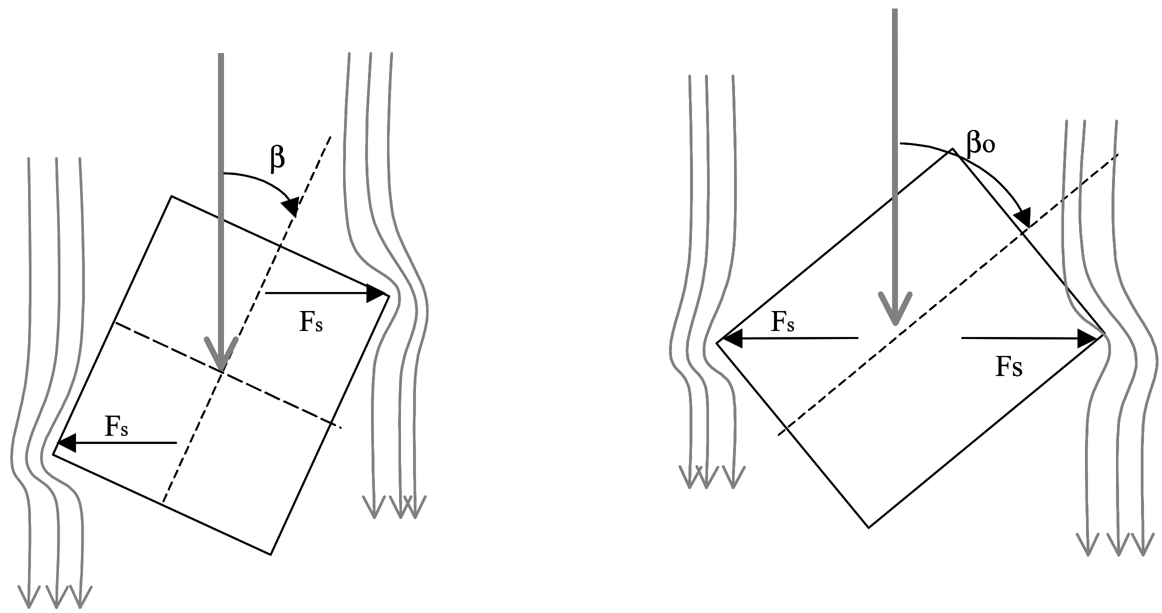



Figure 1. Bluff Body Un-separated Flow Aerodynamic Forces

The angle of zero yaw moment $(\beta_0)$ can be easily calculated for any box as

$$\beta_0 = \tan^{-1}\left( {}^{length}\!/\!_{Width} \right)$$

The center of pressure for forces due to circulation effects can be taken as the point where the streamlines are deflected the most. In the case of the bluff body this corresponds to the outmost corners of the box. The forces affecting the yaw motion (Fs) can be broken down into two components Fx and Fy along and perpendicular to the side of the box with moment arms of ${}^{w}\!/\!_{2}$ and ${}^{l}\!/\!_{2}$ respectively.

$$F_X = F_S \times Sin(2\beta + \beta_0)$$
$$F_Y = F_S \times Cos(2\beta + \beta_0)$$

where the term $(2\beta + \beta_0)$ accounts for the angle of zero yaw moment being greater for a longer box.

The total moment can then be calculated for any arbitrary box as

$$YM = 2 \;\; F_S \;\; \left[ \left( \frac{l}{2} \; - \frac{w}{2} \right) \;\; Sin\left(2\beta + \beta_0\right) \;\; Cos\left(2\beta + \beta_0\right) \right]$$

Using the trigonometric identity $Sin(2\beta) = 2\sin(\beta)\,Cos(\beta)$ and simplifying the total yaw moment becomes

$$YM = F_S \;\; Sin\left(4\beta + 2\beta_0\right) \;\; \left( \frac{l}{2} - \frac{w}{2} \right)$$

In addition wind tunnel studies [Ref 5] showed significant flow separation due to the sharp corners on bluff bodies. This flow separation creates bubbles of low pressure, which will produce lateral forces perpendicular to the sides of the box affecting the yaw moment of the box. These forces can be modeled as shown below.

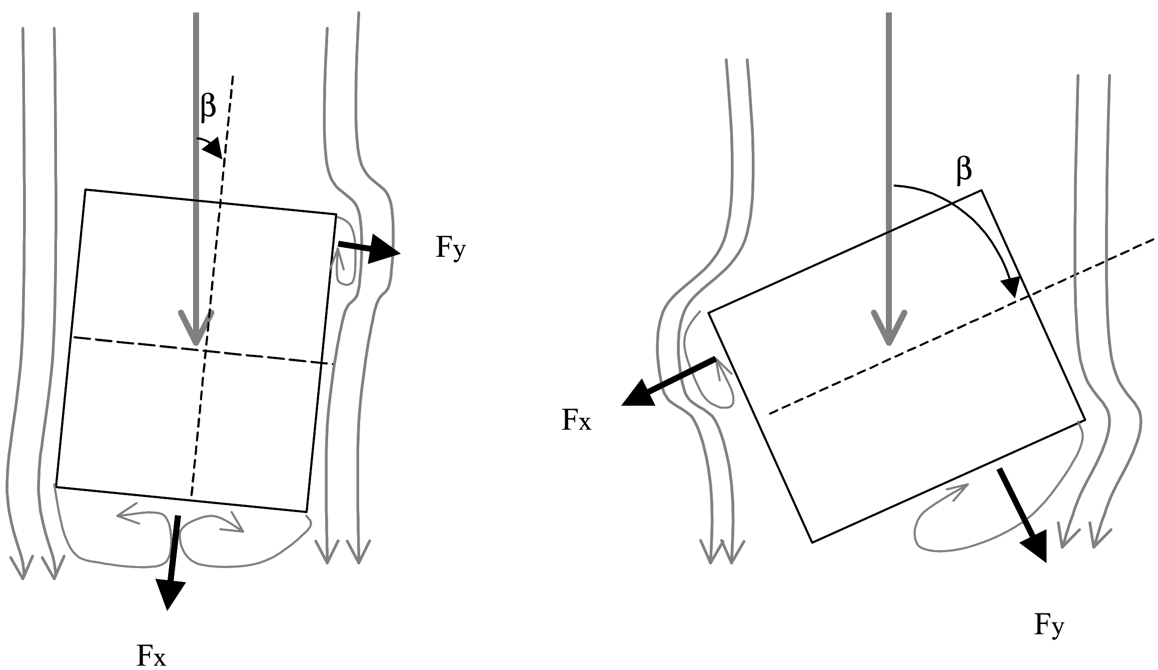



Figure 2. Bluff Body Separated Flow Aerodynamic Forces

The separation bubble produces a difference in pressure between opposite sides of the box resulting in a force along the body x-axis (Fx) which will be at a maximum at β=0 and go to zero at β=90°. It can be represented as

$$F_X = C_P \ Area = C_P \ h \ w \ Cos(\beta)$$

Similarly the force along the body y-axis can be calculated as

$$F_Y = C_P \ h \ l \ Cos(\beta)$$

Since these forces are caused by pressure forces they are perpendicular to the box surface and the center of pressure is located at the center of the flow separation bubble. The yaw moment caused by the flow separation is

$$Y_M = \left[F_B \ \ l \ \ Sin(\beta)\right] \left[{}^{l}\!/_{2} \times Cos(\beta)\right] - \left[F_B \ \ w \ \ Cos(\beta)\right] \left[{}^{w}\!/_{2} \ \ Sin(\beta)\right]$$

*Where*

$$F_{\mathbf{B}} = C_P \ H$$

This simplifies to

$$Y_M = F_B \left[{}^{l^2}\!/_{4} \ Sin(2\beta) - {}^{w^2}\!/_{4} \ Sin(2\beta)\right]$$

Combining the two results gives a total yaw moment of

$$\begin{aligned} Y_M = F_S & \left[{}^{l}\!/_{2} \ Sin(4\beta + 2\beta_0) - {}^{w}\!/_{2} \ Sin(4\beta + 2\beta_0)\right] + \\ F_S & \left[{}^{l^2}\!/_{4} \ Sin(2\beta) - {}^{w^2}\!/_{4} \ Sin(2\beta)\right] \end{aligned}$$

The magnitude of the constant $F_S$ was tuned for a best match of the wind tunnel data and was plotted against the CONEX wind tunnel data obtained by reference (14) in figure (3). The theoretical mathematical model matches the test data with the only noticeably difference being at the 45 and 135 degree points. At these points the wind tunnel data shows a sharp jump, most probably due to the corrugated surface and hardware on the CONEX box.

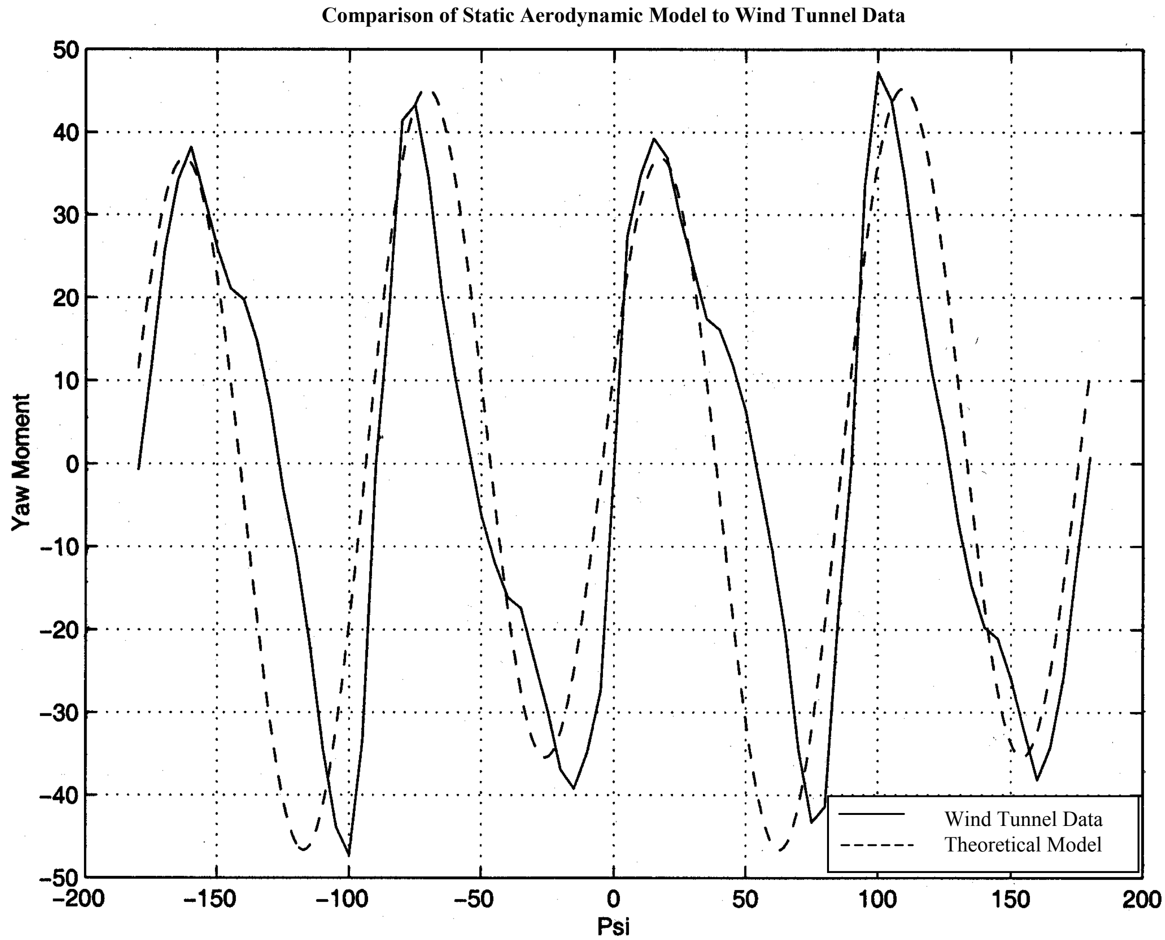



Figure 3. Technion Wind Tunnel Data vs. Theoretical Model

Of particular note is that when integrated over a 360-degree rotation, the net yaw moment produced by the steady aerodynamics is zero. If used in a simple yaw degree of freedom model [Appendix A] the steady aerodynamics will not produce the motions seen in flight. This provides evidence that unsteady aerodynamic forces are required to account for the bluff body load motions seen in flight. This is consistent with the findings by Watkins and Prabakhar [Ref 18,3] that unsteady aerodynamics and coupled yaw-pendulum modes account for the instabilities of cargo containers.

The Technion wind tunnel study provided accurate and comprehensive measurements of the six components of the static aerodynamics of the CONEX as functions of angle of attach and sideslip [Ref 15]. These were implemented in a subroutine as tables of data every 5 degrees along with a lookup cone. Details are given in appendix (B).

### 3. Unsteady Aerodynamics Analysis

The work in reference (5) included an unsteady forced oscillation test. In this test it was noted that the bubble development had a phase lag associated with it though asymmetries in the pressures in the bubbles indicated that unsteady effects other than simple phase lag exist. Unfortunately no data was available on the phase lag and as such could not be mathematically modeled.

Another well known unsteady effect of bluff bodies is that of vortex shedding. In his book on aeroelasticity [Ref 19] Y.C. Fung addresses vortex shedding and notes that a satisfactory theory on vortex shedding has yet to be developed. Nor have the experiments been comprehensive enough to permit useful generalizations. It is known that vortex shedding causes impulsive lateral forces and corresponding torsional moments, which depend on the wind speed. In the case of the external bluff body this could well explain the coupling seen between the lateral pendulous motions and yaw.

A theoretical model of vortex shedding is unknown, but its effects on the load yaw motions in flight are known. Using this knowledge an approximate yaw moment correction can be introduced which represents the vortex shedding plus a bubble development phase delay reproducing the flight yaw motions.

To generate an unsteady aerodynamics model it was assumed that a force was being developed on the downwind sides of the box and was a function of the relative wind velocity. As the box starts to turn, the relative winds on the opposing sides of the box are the same. As the yaw rate increases the relative wind on one side of the box will increase while the opposite side will experience a corresponding decrease s as shown in figure (4).

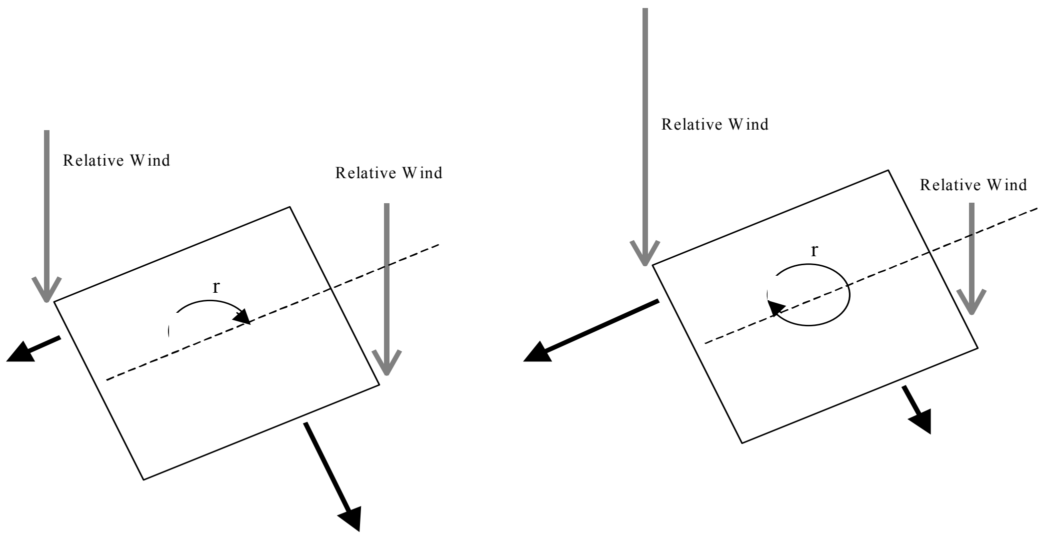



Figure 4. Relative Wind Model

In order to produce the motions seen in flight using a swiveled sling connection, the initial force in the direction of spin has to overpower the force opposing the spin resulting in a net positive yaw moment. As the box yaw rate accelerates the relative wind of the force producing the positive yaw rate decreases and the relative wind of the force opposing the spin increases. This reduces the yaw moment until a balance is found and the box stabilizes in a steady state spin.

These observations were used as a basis for an empirical model of the dynamic aerodynamics forces.

$$Y_M = K_{Dyn} \, {}^{q}\!/_{Va} \, Sign(r)\left(Va - K_g r\right)$$

Where $K_{Dyn}$ accounts for the various unknown constants and $K_g$ is a conversion factor from rotational speed of the box corner to linear speed in the relative air direction. The factor $Sign(r)$ causes the yaw moment to act in the direction of rotation and the factor $\left(Va - K_g \, r\right)$ accounts for the effect of relative wind.

## D. YAW RESISTANCE AT THE HOOK

A single yaw degree of freedom simulation model [Appendix A] was used to model yaw moment corrections for the GenHel / Slung Load program. The corrections for forward flight yaw motions focused on two aspects, the yaw motions due to dynamic aerodynamics and the yaw resistance at the hook. Yaw motions in hover are due to the effect of rotor down-wash swirl on the load

Two different sling configurations were used in the flight tests. One configuration fixed the clevis of the sling directly to the hook on the helicopter and the other used a swivel between the sling and helicopter. In the absence of a swivel the sling legs periodically windup and unwind due to the combination of yaw moment due to unsteady aerodynamics and the yaw resistance at the hook. With a swivel, the load spins up to a steady yaw rate determined by the combined unsteady aerodynamics and swivel friction. These two resistance phenomena were modeled separately as followed

Yaw resistance at the hook without a swivel was determined in tests conducted at the Ames Mechanical Engineering Laboratory. The CONEX box was suspended from a crane with a standard four-legged military sling rated at ten thousand pounds and the crane hook secured against. A motor was connected to the bottom of the CONEX that provided readout of the torque required to rotate the load. Measurements of torque and height were taken at regular rotation intervals up to ten turns. Ten turns was the maximum number of revolutions seen in flight test at 70 knots. The resulting sling geometry at ten turns is seen in figure (5) and numerical results are plotted in figures (6) and (7). The height variation with rotation angle [Fig 6] shows a total shortening of only of 21 inches for the total sling length. However the corresponding sling geometry [Fig 5] shows about one half of the total sling length is involved in the windup.

The plot of torque vs. rotation [Fig 7] shows two distinct regions of behavior. There is nonlinear behavior in the startup region from 0 to 270 degrees and a linear region after that. The measurements also show significant hysteresis, but the source of this was not identified. A table look-up program [Appendix C] gave hook yaw resistance due to sling windup with data taken from the average of the two hysteresis loops in figure (7).

Yaw resistance due to swivel friction was modeled as

$$YM_{Swivel} = -Kswivel \quad r$$

$$where$$

$$Kswivel = 1.5 \; {ft \quad lb} \Big/ {\frac{rad}{Sec}}$$

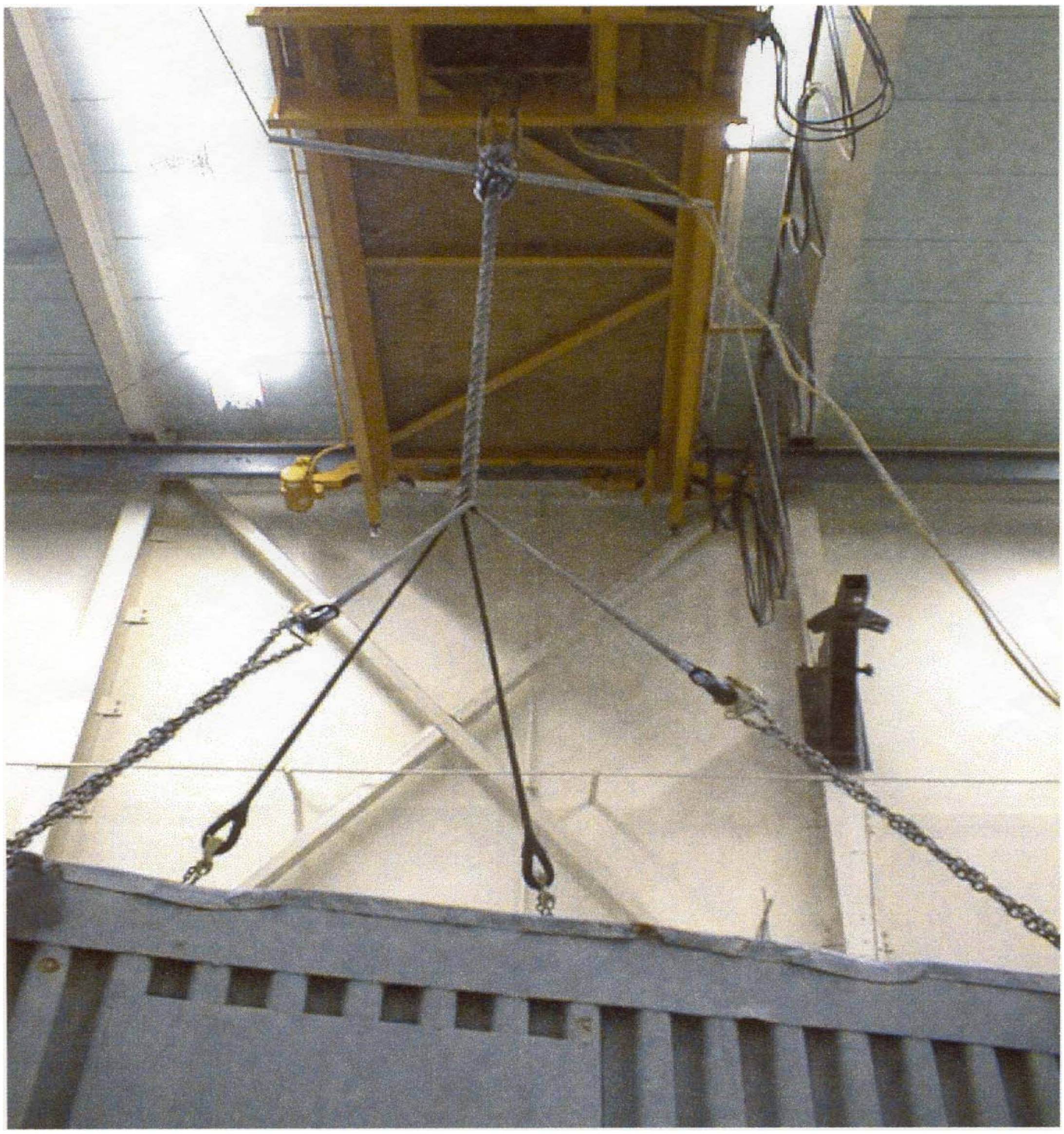


Figure 5. Sling Wind-Up Test at 10 Revolutions

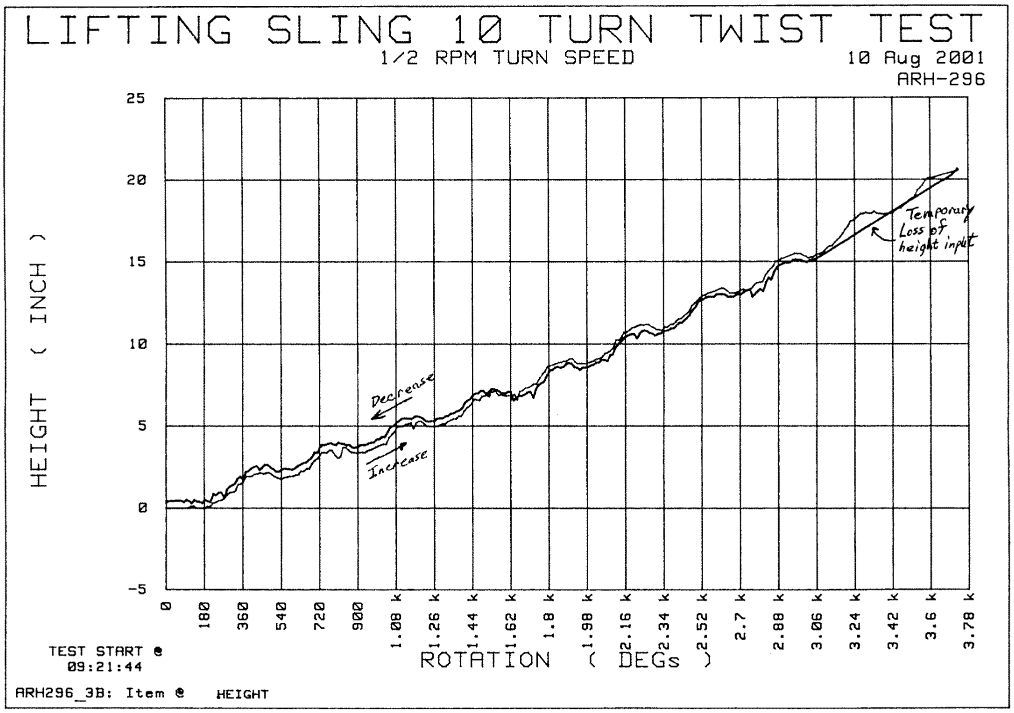



Figure 6. Height vs. Rotation

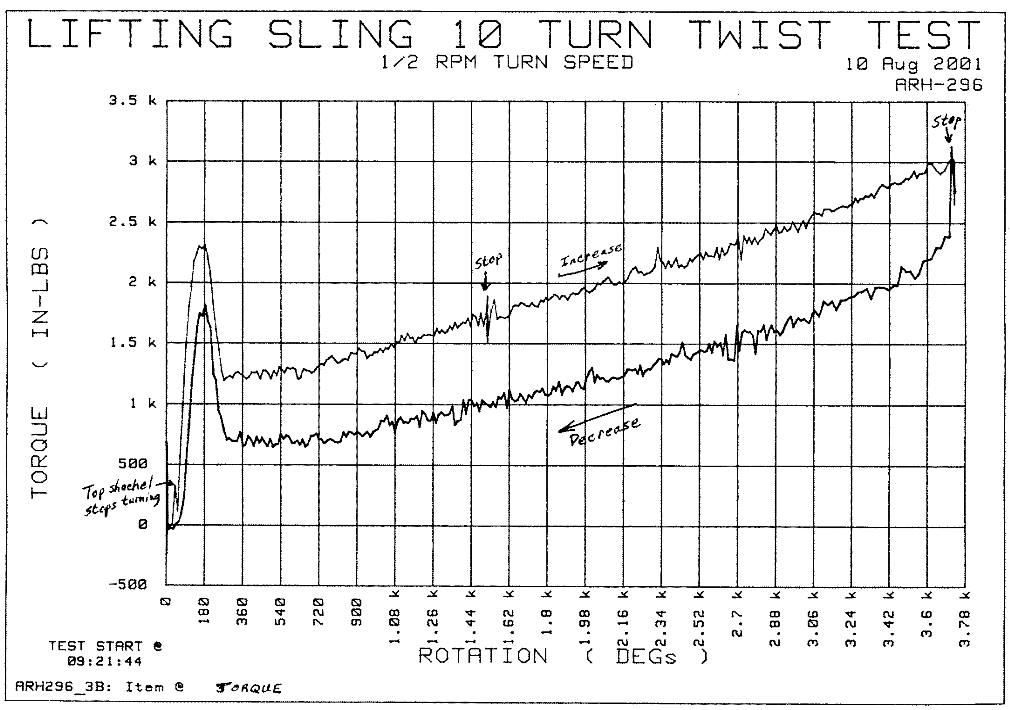



Figure 7. Torque vs. Rotation

### 1. Hover Corrections

It is assumed that the load is entirely immersed in the helicopter wake at hover and that the wake is entirely behind the load at 20 knots forward airspeed. The yaw moment due to wake swirl is therefore modeled by the following linear equation.

$$YM_{Swirl} = Kswirl \times Max\left(0,\ 1 - {}^{Va}\!/_{20}\right)$$

where Va is in knots.

The constant, Kswirl, can be determined from the observed steady yaw rate at hover with a swivel (45 degrees per second) and the swivel friction constant. The steady state solution from the one-degree of freedom model is

$$r_{SS} = \frac{Kswirl}{Kswivel}$$

whence Kswirl winds up to be 40 ft lbs of torque.

### 2. Model Validation

The maximum yaw rates for both swivel and unswivel cases were calculated from flight test data. A series of simulations were conducted at varying airspeeds and the simulation maximum steady yaw rate for the swivel and the non-swivel case were plotted against flight test data in figure (8) and (9). A linear fit of the flight test data was plotted for comparison. As expected yaw rates for the swivel case was noticeably higher than for the unswiveled case for all airspeeds.

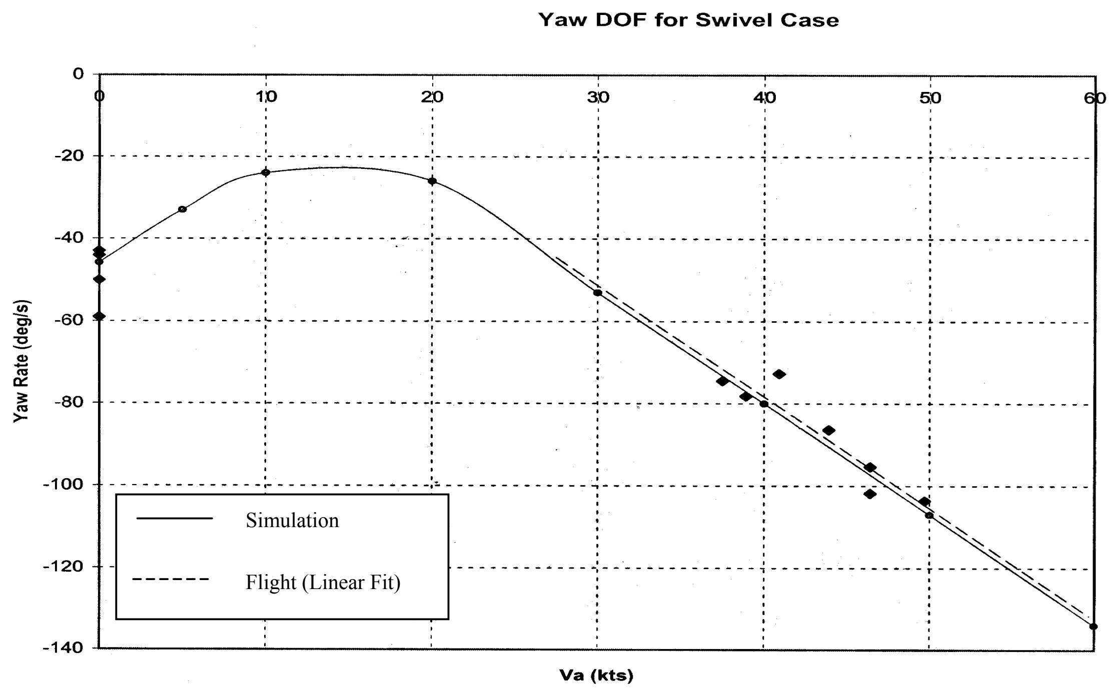



Figure 8. Yaw Motion Corrections Verification, Swivel Case

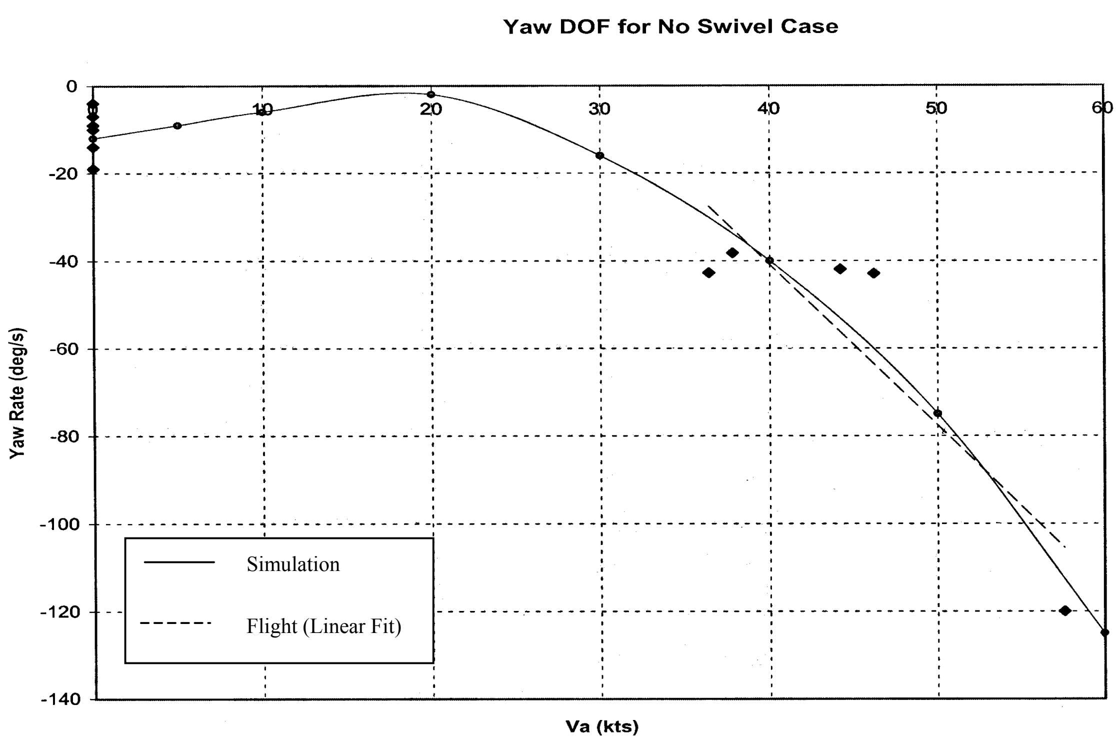



Figure 9. Yaw Motion Corrections Verification, Swivel Case

The simulation model constants $K_{Swivel}$, $K_{Dynam}$ and $K_{Swirl}$ were tuned to flight test data at 40 knots for the unswiveled case and a value of .3 ft-lbs for $K_{Dynam}$ was obtained.

The equations were included into the GenHel / slung load module [Appendix J]. The simulation was then ran using control inputs from flight test data [Appendix D] and compared to flight tests. Figure (10) shows a typical result for a swivel and no-swivel case and more comprehensive charts are shown in appendix (D). The simulations conducted for both the swiveled and unswiveled cases accurately modeled the gross behavior seen in flight. Higher frequency behavior related to detailed variations during each revolution is not as well simulated but has the same frequency content.

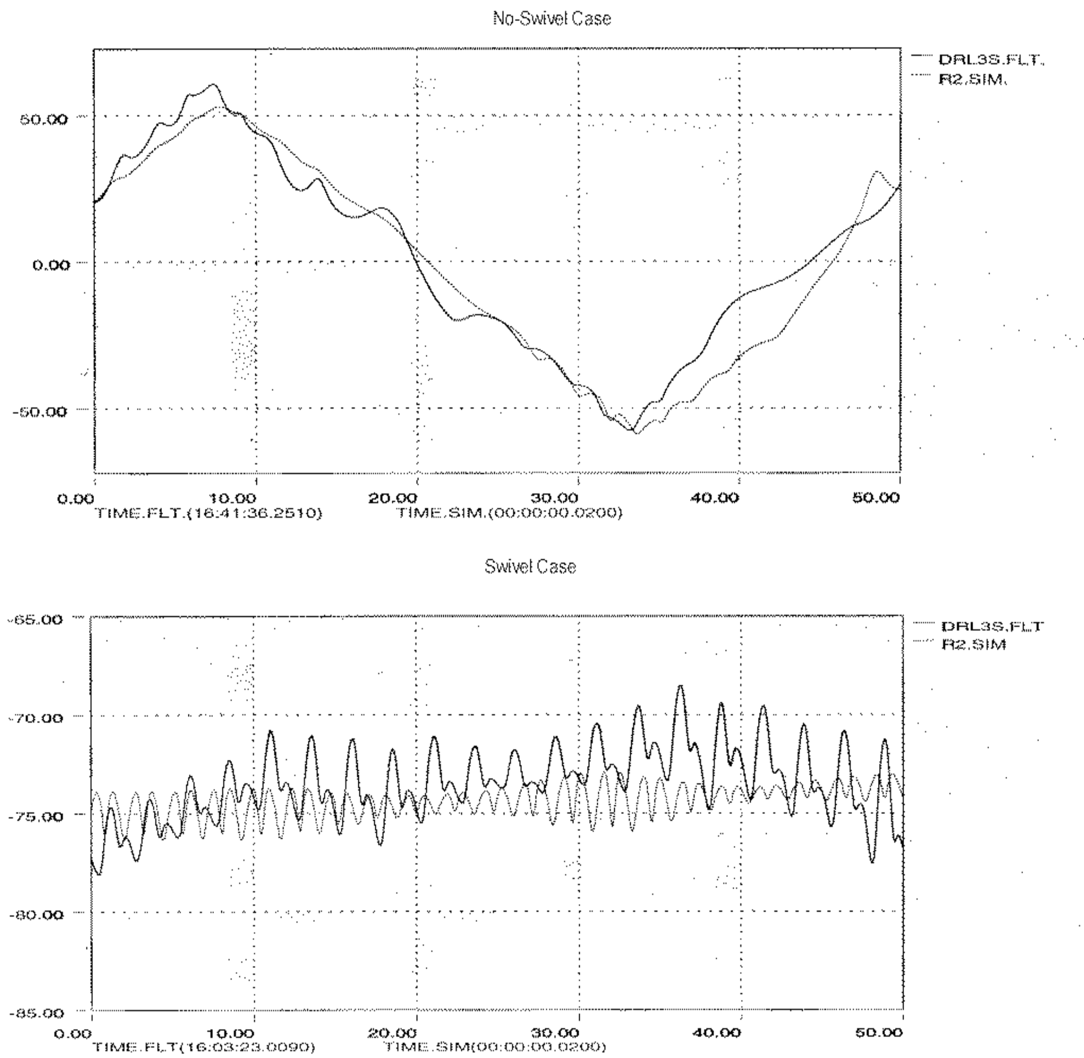



Figure 10. Yaw Rate Validation Plots

# III. STABILIZATION DEVICE

## A. OVERVIEW

This section considers the principal categories of unstable loads and corresponding modes of instability, the general design considerations for stabilization devices, and the available stabilization concepts and their operational suitability. The controllable fin stabilizer assembly is the concept selected for this study.

## B. MODES OF INSTABILITY

Bluff body research [Ref 12, 21] showed a lateral to directional-coupled unstable pendulum as the major source of instability. These loads would benefit most from a stabilization system that controls the yaw motions of the load.

Flat plate loads exhibit aerodynamic instability in which pitch is always strongly involved in the instable motion [Ref 23,24]. These loads require a stabilization system that controls pitch as well as yaw.

Aerodynamically active loads are the most unstable type of loads. These loads and their aerodynamic surfaces can generate significant forces and moments. Analyzing their instability as loads as analyzing their instability as a rigid body aircraft, that is the instability can be determined from linear analysis and ordinary stability derivatives. A stabilization system for these types of loads would have to be capable of controlling the inherent forces and moments of the loads.

## C. DESIGN CONSIDERATIONS

Operational suitability of the stabilization device requires that they be safe, imposing a minimal weight penalty and be easy to use.

Safety of the aircraft, crew and load handlers are of utmost importance. A stabilization device may introduce its own hazards, such as the possibility of a drag chute becoming entangled in the tail rotor, or the possibility of blowing the load about in the downwash of heavy helicopters, endangering ground personnel.

The weight of the stabilization device reduces the payload capacity. Ease of use relates to ease of transportation, assembly and attachment with minimal support and special equipment requirements. This is affected by weight, size and complexity of the device.

## D. APPLICATION

Any stabilization system developed for operational use must be applicable to a wide range of aerodynamically unstable loads. Flow separators and stabilization systems mounted directly onto the load are very load specific and can be difficult or impossible to transfer to other similar loads. For example a stabilization system designed to fit on a CONEX box might not be adaptable to a vehicle displaying similar aerodynamic instabilities. The multi-service Helicopter External Air Transport manuals (HEAT) [Ref 25] have detailed information on numerous loads carried by U.S. military helicopters. A review of the loads authorized for transport with a single point suspension system gives an idea of the range of load sizes that could benefit from a stabilization system. A review of the loads authorized for transport with a single point suspension system gives an idea of the range of load sizes and max operational speed for loads that could benefit from a stabilization system.

| Load | Load Type | Distance Between Lift Points (ft) | Max Airspeed (Kts) |
|---|---|---|---|
| HAWK Acquisition Radar | Bluff Body | 7 X 4 | 75 |
| Fuel Dispensing System | Bluff Body | 8 X 4 | 60 |
| 5-Ton Truck | Bluff Body | 20 X 8 | 80 |
| 1 ½-Ton Trailer | Bluff Body | 10 X 5 | 45 |
| Water Trailer | Bluff Body | 8 X 5 | 80 |
| Collapsible Tank Assembly | Bluff Body | 15 X 8 | 35 |
| Aircraft airmobile maintenance shop | Bluff Body | 15 X 10 | 60 |
| CONEX Box | Bluff Body | 8 X 6 | 60 |
| Medium Span Bridge | Flat Plate | 20 X 15 | 60 |

| | | | |
|---|---|---|---|
| Floating Bridge Span | Flat Plate | 15 X 15 | 50 |
| Flatbed Trailer | Flat Plate | 25 X 15 | 70 |

Table 1. Load Parameters

## E. SELECTED STABILIZATION DEVICE

The merits of several load stabilization systems are compared in table (2).

| Stabilization Device | Advantages | Disadvantages |
|---|---|---|
| Active Jet Controls | No interaction with load handlers | Weight, Complexity |
| Spinning Wheel | | Weight, Complexity |
| Flow Diverters | Simplicity, Weight | Load Specific |
| Wake Splitters | Simplicity | Load Specific |
| Active Arm Controller | Inherent to Helo | Weight, Complexity |
| Controllable Fins | Simplicity, Weight | Attached by Ground Crew |

Table 2. Stabilization System Comparison

The theoretical stabilization system selected for this study is a tail fin assembly attached to a frame. An active control surface along the training edge of the airfoil is proposed. This system is attached by short links between the load and sling as shown in figure (11). This system is the most promising to provide a simple, low-weight system that would be feasible for military and civilian operations and allow for structural adjustments to make it applicable to a wide range of loads of various design.

Figure 11. Load Stabilization System

# IV. STABILIZATION PROGRAM DEVELOPMENT

Equations for the stabilizer aerodynamics and simulation model code are given in this chapter. The aerodynamics is based on those given in reference (20) for isolated control surfaces. The model was first implemented in a MATLAB program to provide illustrative results and confirm the equations. They were subsequently implemented in FORTRAN code and integrated into the GenHel / Slung Load simulation.

## A. FLAT PLATE AERODYNAMICS

Tischler [Ref 20] conducted an in-depth analysis of flat plate aerodynamics and the key points are summarized here

Tischler found that the forces on flat plates were primarily from distinct circulation and edge vortex effects and could be modeled separately in three regions of the pitch and yaw domain which are shown in figure (12).

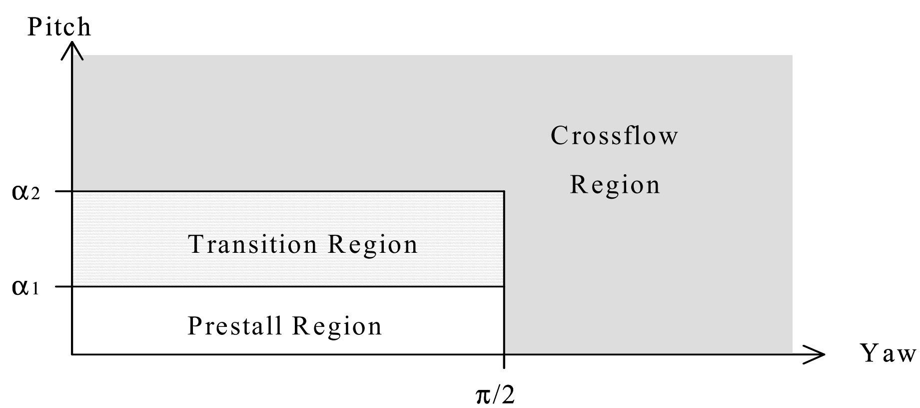



Figure 12. Flat Plate Computational Regions

Aerodynamics in the pre-stall region is based on thin airfoil theory and all forces are primarily due to circulation effects and depend on dynamic pressure (q) and the angle of attack. As in basic thin airfoil theory all forces can be assumed to be acting at the quarter chord point of the plate.

The cross-flow region was that region where the entire down-wind surface of the flat plate is in stall. In this region drag forces predominate and are due primarily to flow separation and vortex effects. In this region the center of pressure is at the geometric center of the plate.

In the transition region the aerodynamic forces are modeled as varying linearly across the region from the pre-stall to the cross-flow regions. The forces in this region are a combination of circulation and edge vortex effects. The center of pressure for this region transitions from the quarter chord to geometric center.

## B. STABILIZER AERODYNAMIC MODEL

The airfoil for a load stabilization system will experience three different flow regimes; pre-stall, stall transition and cross flow as shown in figure (13). The different flow regimes around the airfoil are defined by two stall angles; the angle of attack where the airfoil first starts to enter stall $\left(\alpha^1_{Stall}\right)$ and the angle of attack where the airfoil is unable to produce any lift due to circulation effects $\left(\alpha^2_{Stall}\right)$. In addition the airfoil is considered to be in total stall when the airflow is along the leading edge of the airfoil $\left(\beta = \pm 90^0\right)$ or if reverse flow over the airfoil exists $\left(|\beta| \rangle 90^0\right)$. Table (3) shows the flow regimes in the α β plane

For this model $\alpha^1_{Stall}$ is $20^0$ and $\alpha^2_{Stall}$ is $45^0$ . α is the pitch angle between the x-axis of the load and the free stream velocity vector and β is the yaw angle between the x-axis of the load and the free stream velocity vector.

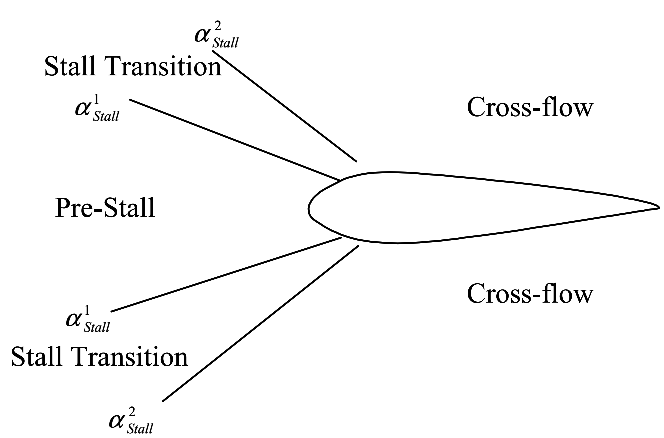



Figure 13. Flow Regimes for Load Stabilization Airfoil

| | α | β |
|---|---|---|
| Pre-Stall | $\lvert\alpha\rvert \langle \alpha^1_{Stall}$ | $\lvert\beta\rvert \langle \pi/2$ |
| Stall Transition | $\alpha^1_{Stall} \langle \quad \lvert\alpha\rvert \quad \langle \alpha^2_{Stall}$ | $\lvert\beta\rvert \langle \pi/2$ |
| Cross flow | $\lvert\alpha\rvert \rangle \alpha^2_{Stall}$ | $\lvert\beta\rvert \rangle \pi/2$ |

Table 3. Control Surface Flow Regimes

### 1. Symbols

In calculating the forces and moments the following constants and equations are defined.

| | |
|---|---|
| $u, v, w$ | Body axis components of velocity |
| $\alpha, \beta$ | Angle of attack and of yaw respectively |
| $S$ | Stabilizer planform area. |
| $C$ | Stabilizer chord. |
| $C_{Do}$ | Skin friction drag. |
| $C_{Dp}$ | Profile drag coefficient. |
| $C_{Di}$ | Induced drag coefficient. |
| $C_{L\alpha}$ | Lift curve slope. |
| $C_{L_\delta}$ | Coefficient of lift due to control surface deflection. |
| $\delta$ | Control surface deflection |

$$V_{xy} = \sqrt{u^2 + v^2}$$

$$V_{xz} = \sqrt{u^2 + w^2}$$

$$V_{yz} = \sqrt{v^2 + w^2}$$

### 2. Pre-Stall

The airfoil body axis forces $F_X$, $F_Y$ and $F_Z$ for this region are given from the lift and drag coefficients as;

$$F_X = -\tfrac{1}{2}\,\rho\,V_{xy}\;u\;S\left(C_{Do} + \frac{C_{L\alpha}^2\,\alpha^2}{\pi\;e\;AR}\right)$$

$$F_Y = -\tfrac{1}{2}\,\rho\,V_{yz}\;v\;S\;C_{Do}$$

$$F_Z = -\tfrac{1}{2}\,\rho\;V_{xz}^2\;S\left[C_{L\alpha}\,\alpha + C_{L\delta}\;\delta\right]$$

In the prestall region the resultant lift force due to circulation is normally considered to act through the quarter chord of the airfoil. This results in a pitching moment about the geometric center of

$$M = F_Z\,{}^{Ch}\!/_{4}$$

### 3. Cross Flow

In the cross flow regime the stabilizer is completely stalled and all forces are a function of drag. $F_X$ and $F_Y$ are due to skin friction drag and $F_Z$ is a due to profile drag.

$$F_X = -\tfrac{1}{2}\,\rho\,V_{xy}\;u\;S\;C_{Do}$$

$$F_Y = -\tfrac{1}{2}\,\rho\,V_{yz}\;v\;S\;C_{Do}$$

$$F_Z = -\tfrac{1}{2}\,\rho\,V_{xy}\;w\;S\;C_{Dp}\;\left|\sin(\alpha)\right|$$

where $S\left|\sin(\alpha)\right|$ is the frontal area presented to the free stream.

In the cross flow regime the center of pressure coincides with the geometric center so that all moments about the geometric center are zero.

### 4. Stall Transition

Force calculations for $F_X$ in the stall transition regime is a linear interpolation between the forces at $\alpha_{Stall}^1$ and $\alpha_{Stall}^2$.

$$F_X = F_X^1 + \left\{ \left[ \frac{F_X^2 - F_X^1}{\alpha_{Stall}^2 - \alpha_{Stall}^1} \right] \times \left[ |\alpha| - \alpha_{Stall}^1 \right] \right\}$$

where $F_X^1$ is calculated using the pre-stall equation at $\alpha_{Stall}^1$

$$F_X^1 = -\tfrac{1}{2}\, \rho\, V_{xY}\ u\ S \left( C_{Do} + \frac{C_{L\alpha}^2 \left(\alpha_{Stall}^1\right)^2}{\pi\ e\ AR} \right)$$

and $F_X^2$ is calculated using an adjusted equation for the dynamic pressure to the relative span to obtain the correct wind velocity and sign at $\alpha_{Stall}^2$ and is

$$\begin{aligned} &F_X^2 = -\tfrac{1}{2}\, \rho\, V_{xy2}\ u_2\ S\ C_{Do} \\ &Where \\ &\quad u_2 = V_{xz}\ Cos\left(\alpha_{Stall}^2\right) Sign\left(Cos\left(\alpha_{Stall}^2\right)\right) \\ &\quad V_{xy2} = \sqrt{u_2^2 + v^2} \end{aligned}$$

No interpolation is needed for $F_Y$ since its equation is the same for all regimes.

$$F_Y = -\tfrac{1}{2}\, \rho\, V_{yz}\ v\, S\ \ C_{Do}$$

The calculations of $F_Z$ is similar to that for $F_X$.

$$F_Z = F_Z^1 + \left\{ \left[ \frac{F_Z^2 - F_Z^1}{\alpha_{Stall}^2 - \alpha_{Stall}^1} \right] \times \left[ |\alpha| - \alpha_{Stall}^1 \right] \right\}$$

where $F_Z^1$ is calculated using the pre-stall $F_Z$ equation at $\alpha_{Stall}^1$.

$$F_Z^1 = -\tfrac{1}{2}\, \rho\, V_{XZ}^2\ S\ \left(C_{L\alpha}\ \alpha_{Stall}^1\ Sign(\alpha) + C_{L\delta}\delta\right)$$

and $F_Z^2$ is calculated using an adjusted equation for the dynamic pressure to the relative span to obtain the correct wind velocity and sign at $\alpha_{Stall}^2$ and is

$$
\begin{aligned}
&F_X^2 = -\tfrac{1}{2}\,\rho\, V_{xz2}\; w_2\; S\, C_{Dp}\; \left|\sin(\alpha)\right| \\
&\textit{Where} \\
&\quad w_2 = V_{xz}\; Sin\left(\alpha_{Stall}^2\right) Sign\left(Sin(\alpha)\right) \\
&\quad V_{xz2} = \sqrt{u^2 + w_2^2}
\end{aligned}
$$

The pitching moment is calculated assuming the resultant lift force point of application migrates from the quarter to half chord proportionally as $\alpha$ increases from $\alpha_{Stall}^1$ to $\alpha_{Stall}^2$.

$$
M = \left[-F_Z\, C/4\right] + \left[\frac{F_Z\, C/4}{\alpha_{Stall}^2 - \alpha_{Stall}^1}\right] \times \left(\left|\alpha\right| - \alpha_{Stall}^1\right)
$$

### 5. Forces and Moments in Load Body Axes

The above equations are given for vector components in the body axes of the stabilizer surface. Body axes of the horizontal stabilizer are assumed to be parallel to the load body axes so no transformation is required. However, the vertical stabilizer body axes are obtained by rotating counter clockwise 90 degrees about the load x-axis. Consequently, velocity components for the vertical stabilizer axes are obtained from the load body axes components as

$$
\begin{pmatrix} u \\ v \\ w \end{pmatrix}_{Vert\ Stab} = \begin{pmatrix} u \\ w \\ -v \end{pmatrix}_{Load}
$$

The total forces from the horizontal and vertical stabilizers in load body axes are:

$$
\begin{aligned}
\Delta F_X &= F_X^H + F_X^V \\
\Delta F_Y &= F_Y^H + F_Z^V \\
\Delta F_Z &= F_Z^H - F_Y^V
\end{aligned}
$$

The contribution of the stabilizer assembly to the load moments about the load geometric center is:

$$\Delta M_X^L = [-F_Y\ \Delta Z + F_Z\ \Delta Y]^H - [F_Y\ \Delta Y + F_Z\ \Delta Z]^V$$
$$\Delta M_Y^L = [F_X\ \Delta Z - F_Z\ \Delta X + M_Y]^H + [F_X\ \Delta Z + F_Y\ \Delta X]^V$$
$$\Delta M_X^L = [-F_X\ \Delta Y + F_Y\ \Delta X]^H + [-F_X\ \Delta Y + F_Z\ \Delta X - M_Y]^V$$

where the superscripts designate the horizontal and vertical stabilizer forces and $\Delta X, \Delta Y, \Delta Z$ are the body axes coordinates of the stabilizer assembly geometric center from the load geometric center.

### 6. Illustrative Results

A MATLAB model [Appendix E] was used to check the force and moment equations. Calculations and plots were made for α between $\pm 90$ degrees and for β between $\pm 180$.

Stabilizer size and orientation for this model was determined by fitting the stabilizer to a CONEX box load. The airfoil used for both the vertical and horizontal stabilizer was a standard NACA 0015 airfoil with typical aerodynamic values chosen as listed in table (4).

| | |
|---|---|
| $C_{Horz}$ (ft) | 2 ft |
| $S_{Horz}$ | 5 ft |
| $\Delta X_{Horz}$ | -5 ft |
| $\Delta Y_{Horz}$ | 0 ft |
| $\Delta Z_{Horz}$ | -4 ft |
| $C_{Vert}$ | 2 ft |
| $S_{Vert}$ | 3 ft |
| $\Delta X_{Vert}$ | -5 ft |
| $\Delta Y_{Vert}$ | 0 ft |
| $\Delta Z_{Vert}$ | -5.5 ft |

| | |
|---|---|
| $C_{L\alpha}$ | .1 (per deg) |
| $C_{Do}$ | .02 |
| $C_{Dp}$ | .4 |
| $\alpha^1_{Stall}$ | 10 deg |
| $\alpha^2_{Stall}$ | 35 deg |

Table 4. Stabilizer Parameters

The Forces and moments of both stabilizer surfaces were plotted individually and checked for expected orientation and magnitude.  Additional plots are shown in appendix (F).

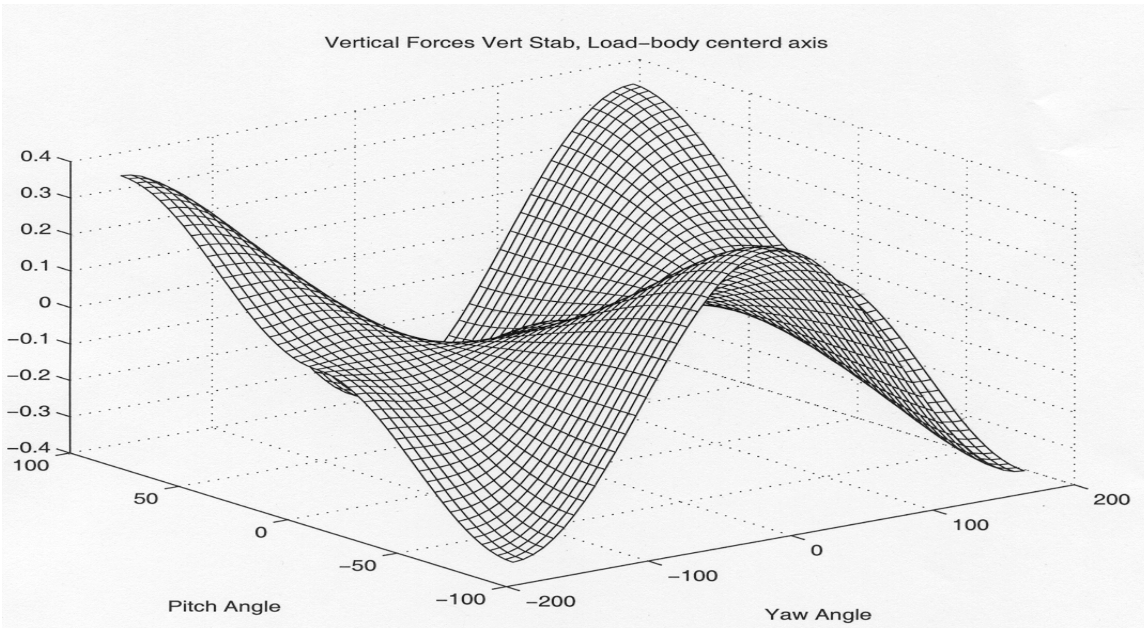



Figure 14.     Vertical Stabilizer Load Yaw Force

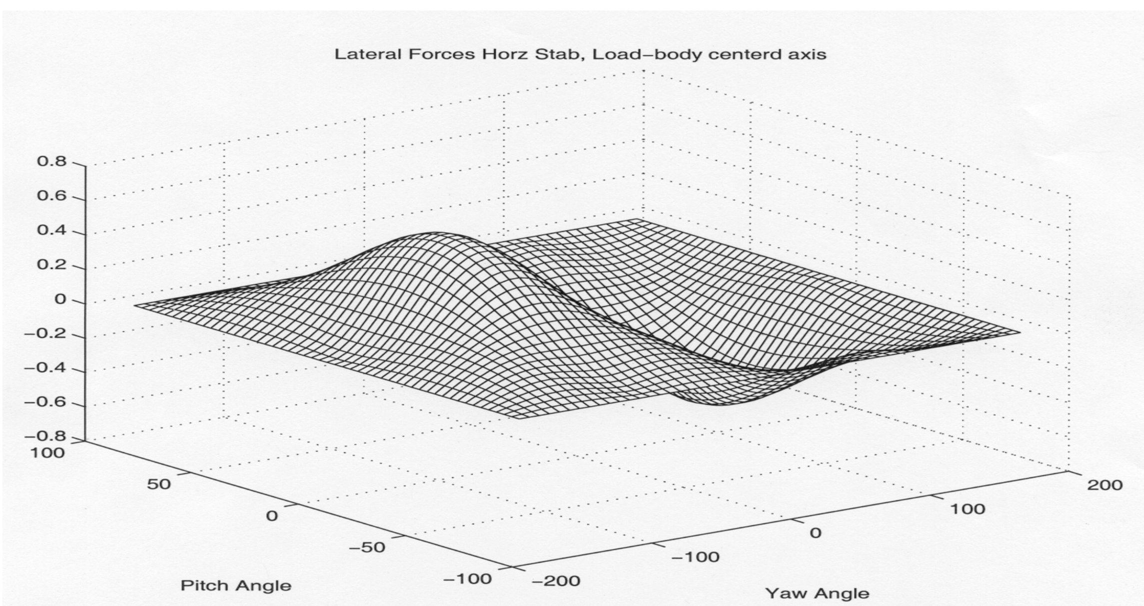



Figure 15.     Horizontal Stabilizer Load Yaw Force

The forces and moments were then transformed from airfoil-centered axis to load body-centered axis and plots of the resultant load-body centered axis forces were checked for expected orientation and magnitude. Additional plots are shown in appendix (H).

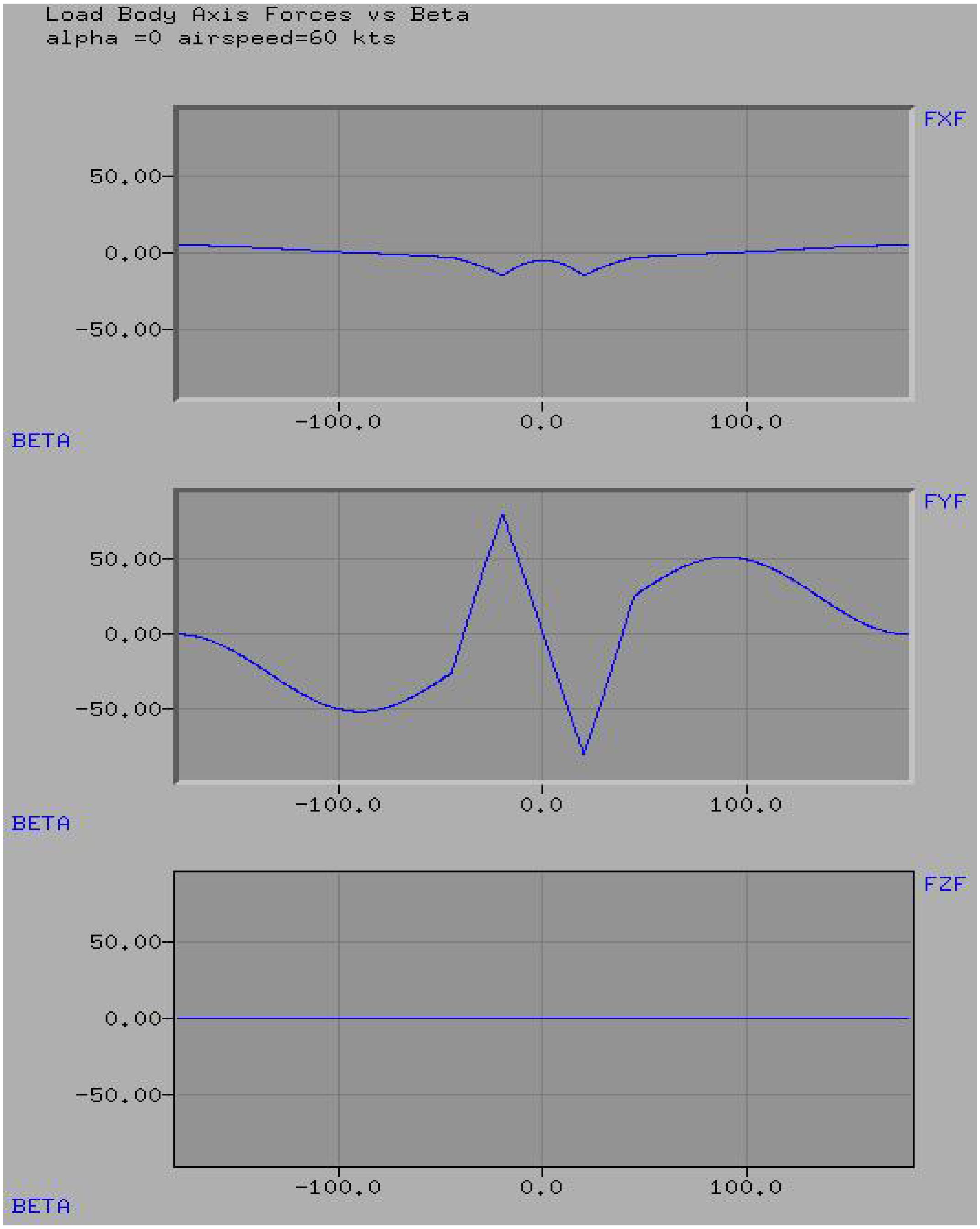



Figure 16. Load Body Axes Forces

## C. FORTRAN MODULE

The above stabilizer equations were implemented in a FORTRAN subroutine [Appendix G] and integrated in the GenHel / Slung Load program. The subroutine models the complete horizontal and vertical stabilizer assembly and generates the stabilizer contribution to the load force and moment sums. Computation of velocity at the stabilizer geometric center must account for load angular velocity.

$$
\begin{aligned}
u &= -va2s2(1) + q \times \Delta Z - r \times \Delta Y \\
v &= -va2s2(2) + r \times \Delta X - p \times \Delta Z \\
w &= -va2s2(3) + p \times \Delta Y - q \times \Delta X
\end{aligned}
$$

where $va2s2$ is the load center of gravity velocity vector and $(p, q, r)$ are the load angular velocities. Data for standard NASA airfoils is available for two-dimensional flow, and $C_{L\alpha}$ can be converted to three-dimensional flow following reference (28).

$$
C_{L\alpha}^{3D} = \frac{C_{L\alpha}^{2D}}{2 + \sqrt{4 + AR^2 \times B^2}}
$$

$$
\begin{aligned}
&\textit{Where} \\
&\quad B^2 = 1 - Mach^2 \\
&\quad AR = \textit{Aspect Ratio}
\end{aligned}
$$

Modifications to the initial GenHel / Slung Load simulation from reference (17) are documented in Appendices J through O. The stabilizer subroutine appears as a single call in the slung load two-body dynamics model [Appendix J], and otherwise common blocks [Appendix O] and input / output subroutines [Appendix K, L] are revised to accommodate the new stabilizer variables. Input values for the stabilizer parameters and options are isolated in a single file given in appendix (I). The post-run processing code was revised as documented in appendix (N).

# V. RESULTS AND CONCLUSIONS

## A. SIMULATION RESULTS

Once the simulation yaw motion had been validated using flight test data a theoretical simulation assessment of the envelope expansion obtained from both passive and active stabilization was conducted. For the assessment only the stability contributions of the vertical stabilizer were considered by setting the area of the horizontal stabilizer to zero. The airfoil chosen for the stabilizer was the NACA 0015 with lift and drag data taken from Ref (28). For the active stabilizer a control surface of 20% of the total chord with a maximum deflection of 20 degrees was used. The stabilizer was placed centered laterally and slightly behind and above the back edge of the box. The corresponding stabilizer geometric center was located at (-8,0,-5) feet in the x, y and z directions respectively. The maximum feasible stabilizer size for use on a CONEX with a 4 legged sling is 4 ft by 5 ft and was the upper limit used in this simulation.

The test case 4,000-pound CONEX box without a stabilizer displays divergent yaw rates above 70 knots in the simulation. The addition of the stabilizer provided damping and extended this range, depending on stabilizer size and whether the stabilizer was passive or active. The effects of the stabilizer were assessed in a series of simulation runs with stabilizers of various sizes over a speed range from 60 to 120 knots. Damping ratio was computed from the yaw rate time history.

$$\zeta = -\frac{\sigma}{\omega_n}$$

*where*

$$\omega_n = \frac{2\pi}{t}$$

$$\sigma = \frac{1}{T}\ln\left(\frac{y_2}{y_1}\right)$$

and the values $T$, $t$, $y_2$ and $y_1$ are obtained from the time history plots.

Figure 17. Damping Ratio Calculations

Additional plots of the yaw rates are given in appendix (P). A damping ratio of .1 was judged to provide adequate yaw motion stabilization and was selected as the target damping.

Figure (18) illustrates the different load yaw motions without and with a 3ft by 4 ft stabilizer. Plotted are yaw angel (PS2) in degrees and yaw rate (R2) in feet per second versus time in seconds. Figure (19) shows the theoretical damping ratio of the passive stabilization device versus airspeed in knots. Figure (20) is for the same size stabilization device, but equipped with a training edge control surface of 20 percent chord.

As seen the CONEX yaw motion and rate at 100 knots demonstrates negative damping without a stabilizer, is damped at .1 with a passive stabilizer and is well damped (.4) for the active stabilizer. A larger area is required for a passive stabilization system to reach the desired damping and will result in undesired additional rolling moment, but does not incur the additional complexity associated with an active stabilization system requiring control systems and rate feedback gyros.

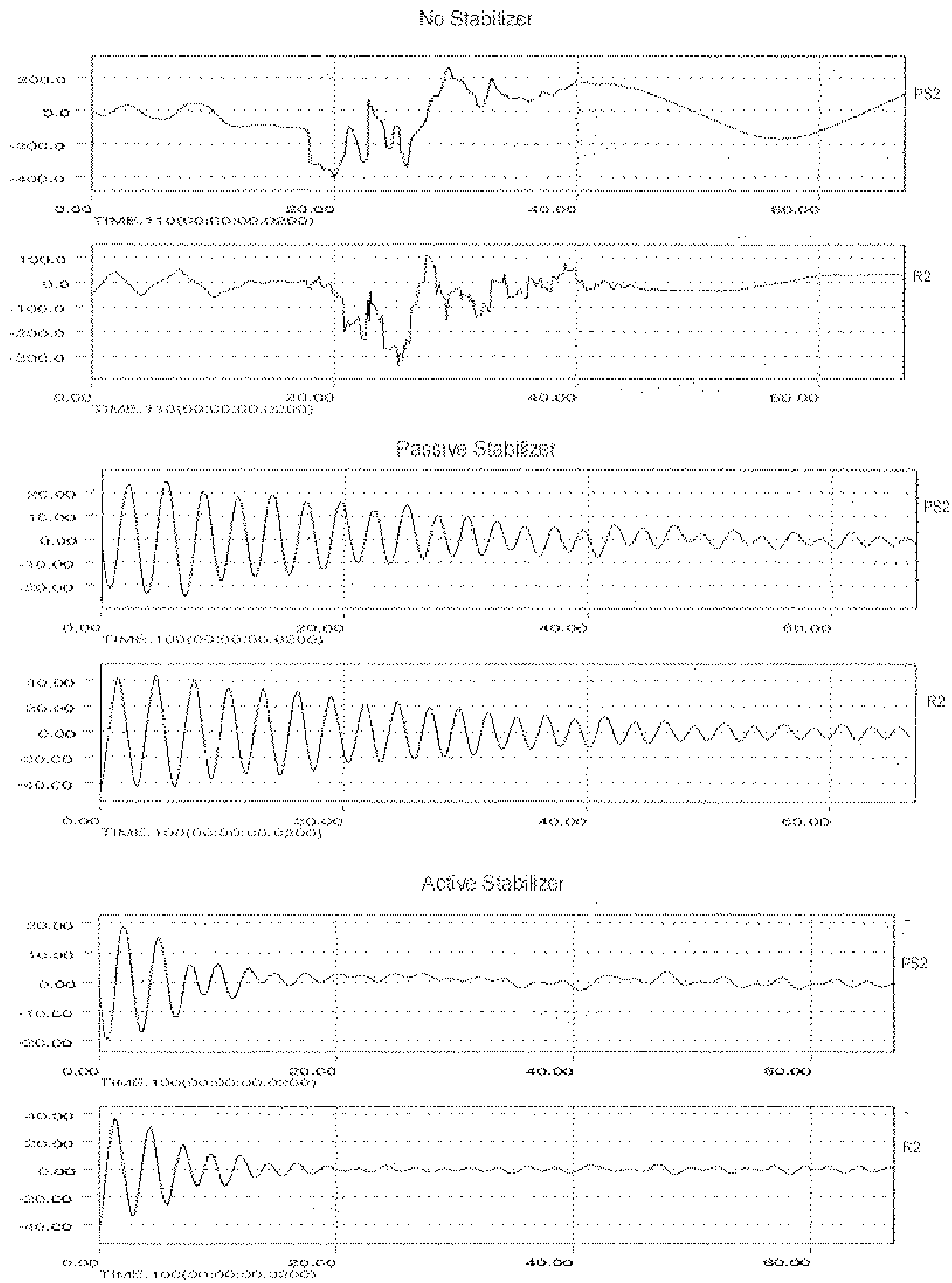



Figure 18. Effect of Stabilizer on Load Yaw Motions

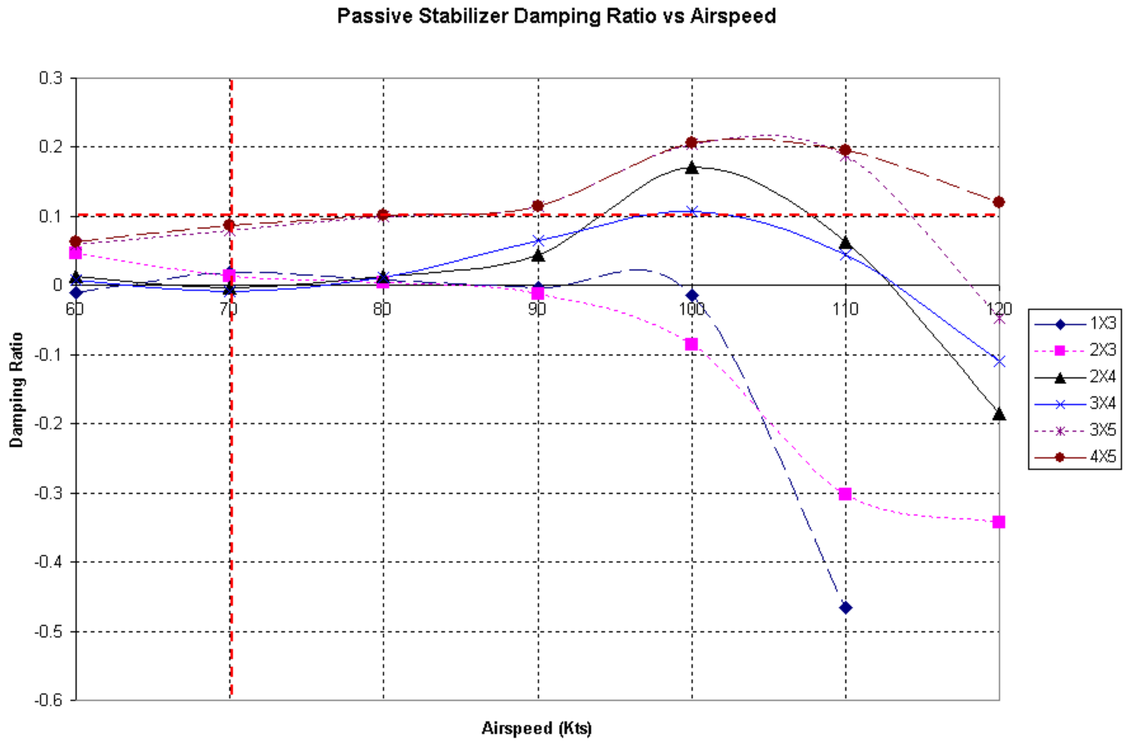



Figure 19. Passive Stabilization System Damping Ratio

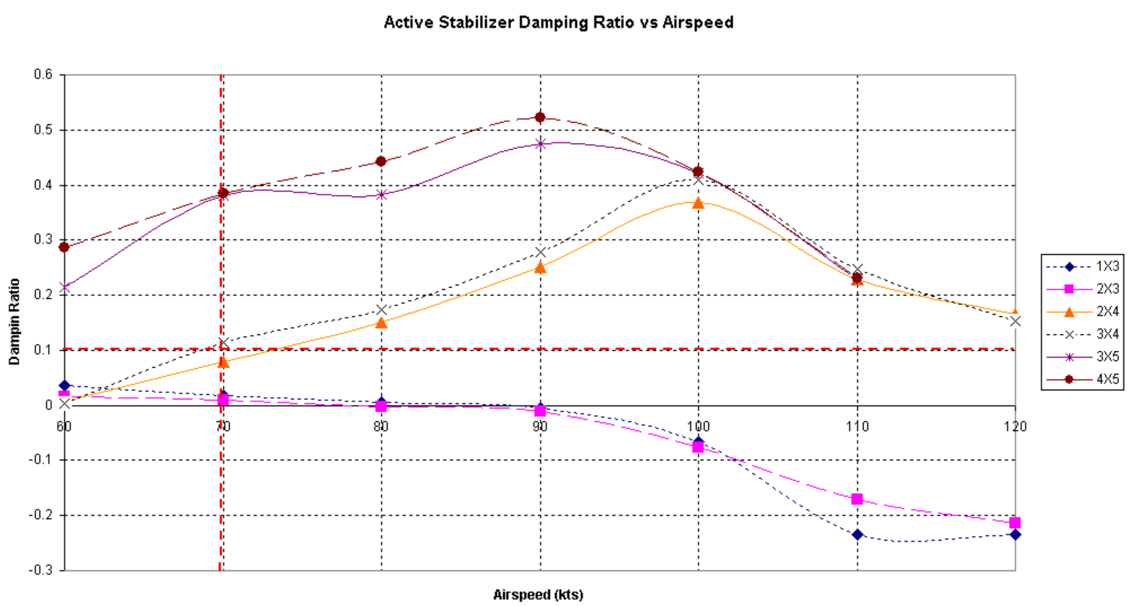



Figure 20. Active Stabilization system Damping Ratio

Results for passive stabilization are shown in figure (19). Passive stabilization with an area of less than eight square feet provided negligible yaw damping, although neutral stability was extended to 90 knots. Stabilizers with fifteen or more square feet provided the desired damping at 70 knots out to 118 knows and beyond. While 8 to 12 square feet areas provided positive damping it only reached the desired damping over a small range of airspeeds.

The addition of active stabilization requires smaller areas as expected [Fig 20] with a requirement for an area of only 8 square feet to produce the desired damping ratio. Stabilizer areas below this provided no improvement over the passive stabilizer. Stabilizer areas of fifteen square feet and up showed large damping, to .52, but little corresponding improvement in yaw rate behavior, as seen in figure (21). Plotted are yaw rate (R2) in feet per second versus time in seconds.

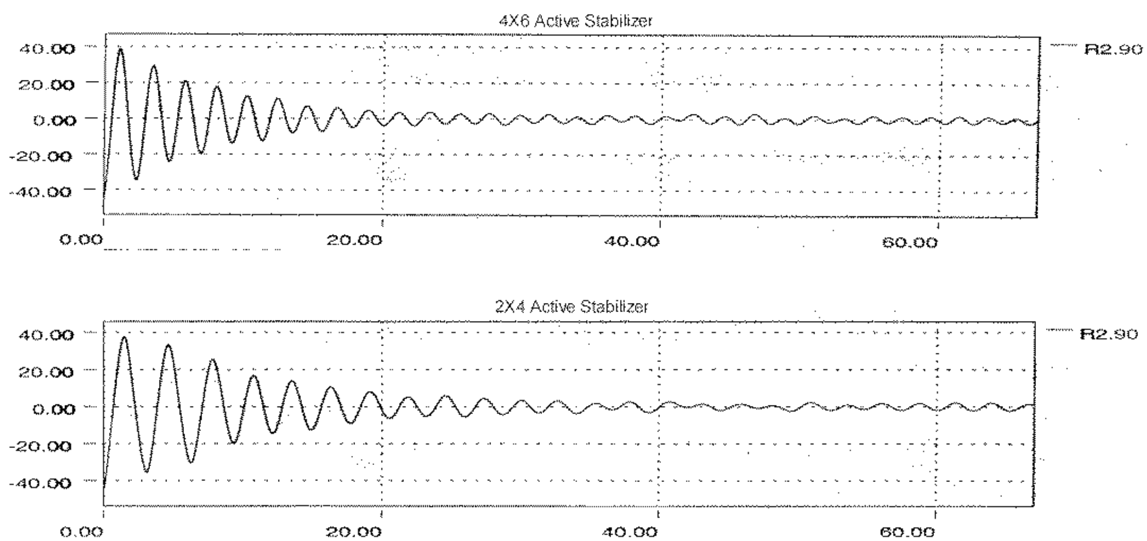



Figure 21. Yaw Rates for Selected Active Stabilization System

Table (6) shows the envelope expansion obtained by the use of passive and active stabilizers. The critical airspeed is that point where the damping of the slung load / stabilization system becomes divergent. The passive stabilizer increases the critical airspeed of the helicopter / slung load system up to 89%. The active stabilization system doubles the critical airspeed.

| Size (ft) | Type | Critical Airspeed (kts) | Percentage Increase |
|---|---|---|---|
| 1X3 | Passive | 85 | 21% |
| 2X3 | Passive | 87 | 24% |
| 2X4 | Passive | 112 | 60% |
| 3X4 | Passive | 113 | 61% |
| 3X5 | Passive | 118 | 69% |
| 4X5 | Passive | 132 | 89% |
| 1X3 | Active | 85 | 21% |
| 2X3 | Active | 87 | 24% |
| 2X4 | Active | 127 | 81% |
| 3X4 | Active | 127 | 81% |
| 3X5 | Active | 135 | 93% |
| 4X5 | Active | 140 | 100% |

Table 5. Speed Envelope Expansion

## B. CONCLUSIONS

The unstable load motions of bluff bodies seen in flight are due to load unsteady aerodynamics. These instabilities raise an important safety concern and can limit the speed envelope for their external transport well below the power-limited airspeed of the helicopter alone. Stabilizer fins provide a promising method of stabilizing bluff bodies both in effectiveness and in operational feasibility.

Results for a 4,000 pound CONEX and a vertical fin stabilizer showed that its 70 knot speed envelope could be extended to 120 knots using a passive stabilizer of fifteen square feet area, or an active stabilizer with eight square feet area and a trailing edge control surface of 20 percent chord. Though a much smaller active stabilization system is needed to obtain the desired damping it comes with the additional complexity of instrumentation and control systems.

Both passive and active stabilization systems displayed a rapid drop in effectiveness at airspeeds below 60 knots, so that additional design analysis would be needed for very difficult bluff body loads with unstable speeds much lower than the CONEX.

## C. FURTHER RESEARCH

Construction and flight-testing of the proposed stabilization system is essential in validating the current simulation. Once validated, application of the stabilization system can be expanded to other types of loads such as flat plate type loads and non box-type bluff bodies like vehicles.

To expand the scope of the stabilization system, a better knowledge of the driving forces due to dynamic aerodynamics are needed. Once these phenomena are understood the simulation can be expanded to cover a wide range of loads allowing the stability and effects of a load to be studied in depth prior to any flight-testing.

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF REFERENCES


1. Chan D., Flower J.W., Simpson A., *Aerodynamically Induced Motions of Bluff Bodies Suspended Beneath Helicopters,* Unites States Army Contract No DAJA 37-73-0477 Final Technical Report, 1975

2. Sheldon D.F., Pryor J., *A Wind Tunnel Investigation of Yaw Instabilities of Box-shaped Loads Underslung from a Helicopter on a Tandem Suspension,* RMCS Technical Note AM/62 R.M.C.S. Shrivenham, U.K., Dec 1974

3. Prabakhar A., *Stability of a Helicopter Carrying an Underslung Load,* European Rotorcraft Forum Paper #16, Aix en Provence, Sep 1977

4. Laub G.H., Kondani H.M., *Wind Tunnel Investigation of Aerodynamic Characteristics of Scale Models of Three Rectangular-Shaped Cargo Containers,* NASA TMX-62, 169, 1972

5. Simpson A., Flower J.W., *Unsteady Aerodynamics of Oscillating Containers and Application to the Problem of Dynamics Stability of Helicopter Underslung Loads,* AGARD-CP-235, May 1978

6. Ronen, T., *Dynamics of a Helicopter With a Sling Load,* PhD Dissertation, Department of Aeronautics and Astronautics, Stanford University, Palo Alto, CA, Aug 1985

7. Smith J.H., Allen E.M., Vensel D., *Design, Fabrication, and Flight Test of the Active Arm External Load Stabilization System for Cargo Handling Helicopters,* Final Report USAAMRDL Technical Report 73-73, Sep 1973

8. Micale E.C., Poli C. *Dynamics of Slung Bodies Utilizing a Rotating Wheel for Stability,* Journal of Aircraft Vol. 10, Dec 1973

9. Pardue M.D., Shaughnessy J.D., *Design and Analysis of an Active Jet Control system for Helicopter Sling Loads,* NASA TP-1397, Jan 1979

10. Liu D.T., *In Flight Stabilization of Externally Slung Helicopter Loads,* USAAMRDL Technical Report 73-5 Contract DAJA 02-70-C-0067, 1973

11. Matheson N., *Stability of Helicopter Slung Loads,* Dept. of Defense, Defense Science and Technology Organization, Aeronautical Research Laboratories, Australia, Aerodynamics Note 364, Aug 1976

12. Gera J., Farmer, S.W. Jr, *A Method of Automatically Stabilizing Helicopter Sling Loads,* NASA TN-D-7593, July 1974

13. Tischler M.B., Shlomit G. ed, *US/Israel Memorandum of Agreement on rotorcraft Aeromechanics and Man/Machine Integration Technology 25th Meeting 27-31 March 2000*, Sep 2000

14. Rosen A., Cecutta S., Yaffe R., *Wind Tunnel Tests of Cube and CONEX Models,* Technion Institute of Technology, Department of Aerospace Engineering , TAE 844, Nov 1999

15. Cicolani, McCoy, Sahai, Tyson, Tischler, Rosen, Tucker, *Flight Test Identification and Simulation of a UH-60A Helicopter and Slung Load,* Journal of the American Helicopter Society, Apr 2001, Also NASA TM-2001-209619, Jan 2001

16. Tyson P, *Simulation Validation and Flight Prediction of UH-60A Black Hawk Helicopter/Slung Load Characteristic,* Naval Postgraduate School, Department of Aeronautics and Astronautics, Monterey Ca., Mar 1999

17. Howlett J., *UH-60A Black Hawk Engineering Simulation Program,* NASA CR-166309, 1981

18. Watkins T.C., Sinacori J.B., Kesler D.F., *Stabilization of Externally Slung Helicopter Loads,* USAAMRDL TR-74-42, Aug 1974

19. Fung Y.C., *An Introduction to the Theory of Aeroelasticity,* New York, Dover Publications, 1969

20. Tischler M.B., *Flight Dynamics Analysis and Simulation of Heavy Lift Airships,* NASA TM-166471, Dec 1982

21. Raz R. Rosen A. Ronen T., *Active Aerodynamic Stabilization of a Helicopter / Sling-Load System,* Journal of Aircraft, Sept 89

22. Hutto A.J., *Flight Test Report of the Heavy Lift Helicopter Flight Control System,* 31st Annual National Forum of the American Helicopter Society, May 1975

23. Matheson N., *The Stability of Portable Bridges Carried on slings Beneath Helicopters,* Dept. of Defenses, Defense Science and Technology Organization, Aeronautical Research Laboratories, Australia, Aerodynamics Report 154, Jan 1980

24. Sheldon D.F., *A Study of the Stability of a Plate-Like Load Towed Beneath a Helicopter,* Journal of Mechanical Engineering Science, Vol 13, 1971

25. *Multiservice Helicopter External Air Transport, Vols I, II, III,* U.S. Army Fm-55-450-3,4,5, Feb 1991

26. Cicolani L.S., Kanning G., *Equations of Motion of Slung-Load Systems, Including Multilift systems,* NASA Technical Paper 3280, Nov 1992

27. Raymer D.P., *Aircraft Design: A conceptual Approach, Third Edition,* AIAA Education Series Air Force Institute of Technology Wright-Patterson Air Force Base, Oh, 1999

28. Jacobs E.N., Ward K.E., Pinkerton R.M., *The Characteristics of 78 Related Airfoil Sections From Tests in the Variable-Density Wind Tunnel,* NACA Report 460

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX A YAWDOF3.M

FORTRAN program to develop and check the correcting equations for the yaw degree of freedom in the GenHel / Slung Load program. Program by Cicolani L. and Ehlers G.

C file yawdof3.f 18 april 01

```
C simulate yaw DOF for swiveled hook, using techdat4 WT data.  current
version is conex4a.f
C revise alfa,beta computation to correspond to load w steady trail
angle rotating about its vertical axis


      program yawdof2

      parameter( nchan = 11)

      real va2s2(3), fa22(3), ma22(3), faw(6), Tbn(3,3)

      real Kr,Kpsi,Fdyn,Izz,ktstfps,dynam,swirl
      real*8 time,dat8(nchan)
      character*20 fn, sn(nchan)
      logical*4 L,openW
      integer flag

      common/xfloat/a(500)/ifixed/ia(50)
      common/lfloat/rload(200),rload3(200),rload4(200),rload5(200)
      equivalence(A(183), rho)
      equivalence(A(359), rtd)

      equivalence(rload5(73), alf2d)
      equivalence(rload5(74), bet2d)
      equivalence(rload5(76), faw(1))
      equivalence(rload5(82), alfao)
      equivalence(rload5(83), betao)

      equivalence(va2s2(1), u)
      equivalence(va2s2(2), v)
      equivalence(va2s2(3), w)


      data
     x  tstop,dt/300., .01/,
     x  vakts,flag/50,0/,
     x  iaero,Aym/1,400/,
     x  sn/
     x 'dr    ','r     ','psi  ','beta  ','alfa  ',
     x 'betao ','alfao','u     ','v     ','w     ',
     x 'ym    '/
```

```
      rho     = .00237689
      rtd     = 57.295780
      dtr     = 1/rtd
      ktstfps = 1.68894
      pi      = 3.14159274

      Kr      = 0
      Kpsi    = 0
      Fdyn    = .7

      thto    = 20
      psio    = 0.
      aoao    = -20.
      Izz     = 1376
      fn      = 'test.xp'
      rdps    = 0.

      namelist/in/tstop,dt,vakts,aoao,Kr,Kpsi,Fdyn,psio,rdps,fn,
     x    iaero, Aym, thto

 103  continue

c  Read in output file name
 90   type *, 'Enter name of output file name.xp'
      read(5,91,err=90) fn
 91   format(A)

c  Read in Va
 92   type *, 'Enter Va'
      read(5,93,err=92)vakts
 93   format(F3.0)


      write(6,in)
 104  type *,'any changes? (CR,1) = (no,yes)'
      read(5,101,err = 104) flag
      if (flag.ne.0) then
         flag = 0
 102     type*,'enter changes to &in'
         read(5,in,err=102)
      end if
 101  format(I4)

      L = openW(1,fn,nchan,sn,'unc3')

      t     = 0
      dto2  = dt/2
      odr   = -60
      r     = rdps*dtr
      o_r   = r
      va    = vakts*ktstfps
      psi   = psio*dtr
      tht   = thto*dtr
      stht  = sin(tht)

      dr    = 0
```

```
 100  continue

      cpsi = cos(psi)
      spsi = sin(psi)

      sph = -stht*spsi
      ph  = asin(sph)
      cph = cos(ph)
      sth = -cpsi*stht/cph
      th  = asin(sth)
      cth = cos(th)

      Tbn(1,1) = CPSI*CTH
      Tbn(1,2) = CTH*SPSI
      Tbn(1,3) = -STH
      Tbn(2,1) = CPSI*SPH*STH-CPH*SPSI
      Tbn(2,2) = SPH*SPSI*STH+CPH*CPSI
      Tbn(2,3) = CTH*SPH
      Tbn(3,1) = CPH*CPSI*STH+SPH*SPSI
      Tbn(3,2) = CPH*SPSI*STH-SPH*CPSI
      Tbn(3,3) = CPH*CTH

      alfa = atan(Tbn(3,1)/Tbn(1,1))
      beta = asin(Tbn(2,1))
      s    = sign(1,Tbn(1,1))
      if (s.lt.0.and.beta.lt.0) beta = -pi - beta
      if (s.lt.0.and.beta.gt.0) beta =  pi - beta

      calf  = cos(alfa)
      salf  = sin(alfa)
      cbet  = cos(beta)
      sbet  = sin(beta)
      u     = Va*calf*cbet
      v     = Va*sbet
      w     = Va*salf*cbet
      q     = .5*rho*va*va

      if (iaero.eq.1) then
         call conex4a(va2s2,fa22,ma22)
         ym   = ma22(3)
      else
         ym = Aym*sin(4*beta)
      end if
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
ccccccc


      dynam= sign(1,r)*q*(va-fdyn*abs(r*rtd))
      swirl= max(0,5*(1-vakts/20))
c      ym   = ym+dynam+swirl

      dr   = (ym-Kr*r-Kpsi*psi)/Izz
      r    = r + (3.*dr - odr)*dto2
      psi  = psi + (o_r + r)*dto2
      t    = t + dt
      odr  = dr
      o_r  = r
```

```
c store unc3 file here.  This is clumsy
      time    = t
      dat8(1) = dr
      dat8(2) = r*rtd
      dat8(3) = psi*rtd
      dat8(4) = beta*rtd
      dat8(5) = alfa*rtd
      dat8(6) = betao
      dat8(7) = alfao
      dat8(8) = u
      dat8(9) = v
      dat8(10)= w
      dat8(11)= ym
      call fwrite(1, time, dat8)

      if (t.le.tstop+dt) go to 100

      call closeW(1)
 301  type *, 'another run? (cr,1) = (no,yes)'
      read(5,101,err=301) flag
      if (flag.ne.0) then
        flag = 0
        goto 103
      end if

      stop
      end


      subroutine serch1(x,tx,nx,ix,sigx)
      dimension tx(1)
      x1 = amax1(tx(1),amin1(x,tx(nx)))
      i = 1
 1    if (i.eq.nx) then
         ix = nx -1
         sigx = 1
      else if (x1.lt.tx(i+1)) then
         ix = i
         sigx = (x1 - tx(i))/(tx(i+1)-tx(i))
      else
         i = i + 1
         go to 1
      end if
      return
      end

c  2-D table lookup routine for an array A(nx,ny).

      function f2d(ix,iy,sigx,sigy,nx,tf)

      dimension tf(1)

      k01 = nx*iy + ix
      k11 = k01 + 1
      k00 = k01 - nx
      k10 = k00 + 1
```

```
f00  = tf(k00)
f01  = tf(k01)
f10  = tf(k10)
f11  = tf(k11)

sigys = 1 - sigx

if (sigy.le.sigys) then
  f2d = f00 + (f10-f00)*sigx + (f01-f00)*sigy
else
  f2d = f11 - (f11-f01)*(1-sigx) - (f11-f10)*(1-sigy)
end if
return
end
```

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX B CONEX4A.F

Forces and Moments data from Technion Institute wind tunnel experiment. Lookup program to extract data for GenHel / Slung Load simulation follows data. Program by Cicolani L>

```
c      subroutine /harrier/tyson/GenHel/batch/sl/conex4a.f

C   conex static aerodynamics from Technion CONEX WT data #4     april
01
C   data revised by techdat4.m to impose
C       (1) doq(aoa,90) = fixed
C       (2) loq, rmoq, pmoq = 0 at (alfa,beta) = (0,90)
C       (3) loq, rmoq, pmoq = 0 at aoa = -90,+90.
C   This version stores tables in (alfa,beta) in [-90,90]x[0,180] and
C   computes (alfa,beta) S.T. |alfa| < 90.
C   PT's wake model omitted this version

      subroutine conex4a
      include 'slvars.cmn'

C123456789C123456789C123456789C123456789C123456789C123456789C123456789C
1
      real alfg_cnx(37), betg_cnx(37),
     x     doqt_cnx(37,37),yoqt_cnx(37,37),loqt_cnx(37,37),
     x     rmoqt_cnx(37,37),pmoqt_cnx(37,37), ymoqt_cnx(37,37),
     x     s(6,2),faw(6),fawo(6),T2w(3,3)

      equivalence(faw(1), fa2w(1))

C fortran stores by column.  each column corresponds to a single value
of beta here - the reverse
C of the storage in milvan_arc.f where each column was a value of alfa.


      data  alfg_cnx/
     x   -90.00,  -85.00,  -80.00,  -75.00,  -70.00,  -65.00,  -60.00,
     x   -55.00,  -50.00,  -45.00,  -40.00,  -35.00,  -30.00,  -25.00,
     x   -20.00,  -15.00,  -10.00,   -5.00,    0.00,    5.00,   10.00,
     x    15.00,   20.00,   25.00,   30.00,   35.00,   40.00,   45.00,
     x    50.00,   55.00,   60.00,   65.00,   70.00,   75.00,   80.00,
     x    85.00,   90.00/
      data  betg_cnx/
     x     0.00,    5.00,   10.00,   15.00,   20.00,   25.00,   30.00,
     x    35.00,   40.00,   45.00,   50.00,   55.00,   60.00,   65.00,
     x    70.00,   75.00,   80.00,   85.00,   90.00,   95.00,  100.00,
     x   105.00,  110.00,  115.00,  120.00,  125.00,  130.00,  135.00,
     x   140.00,  145.00,  150.00,  155.00,  160.00,  165.00,  170.00,
     x   175.00,  180.00/

      data  ((doqt_cnx(i,j), i = 1,37),j=1,10)/
```

```
x    48.33,   50.83,   54.17,   57.85,   62.45,   66.58,   69.82,
x    72.92,   74.55,   76.02,   75.64,   76.02,   74.86,   72.16,
x    68.86,   64.59,   61.22,   59.20,   62.92,   64.43,   64.70,
x    66.15,   68.47,   72.61,   75.98,   78.85,   80.75,   80.24,
x    79.00,   76.56,   73.71,   70.91,   69.08,   65.78,   66.23,
x    65.99,   66.76,
x    45.09,   50.78,   54.36,   57.10,   60.68,   65.12,   68.39,
x    71.51,   73.65,   75.56,   74.92,   75.16,   73.64,   70.98,
x    67.42,   63.77,   61.35,   59.80,   61.13,   64.63,   64.50,
x    65.45,   68.28,   71.94,   75.83,   79.27,   83.05,   82.98,
x    81.48,   80.21,   76.47,   72.34,   69.27,   66.49,   66.39,
x    66.19,   66.87,
x    46.20,   50.93,   54.49,   56.82,   60.07,   63.17,   66.63,
x    70.31,   72.19,   72.89,   73.59,   73.27,   71.04,   68.10,
x    65.61,   62.78,   61.31,   59.39,   59.13,   64.07,   64.40,
x    64.97,   67.26,   71.05,   74.51,   78.81,   82.19,   82.03,
x    81.46,   79.58,   74.98,   70.78,   68.20,   65.89,   66.16,
x    65.17,   66.02,
x    47.74,   52.45,   55.47,   57.01,   59.71,   62.72,   66.67,
x    70.06,   72.24,   73.07,   73.17,   71.49,   69.57,   66.62,
x    64.43,   62.31,   61.48,   60.01,   59.72,   64.65,   65.46,
x    65.38,   67.75,   70.89,   74.03,   76.61,   81.44,   80.64,
x    79.65,   76.86,   73.11,   69.25,   66.92,   65.19,   65.78,
x    64.86,   65.93,
x    50.86,   54.56,   58.32,   59.55,   61.81,   64.17,   66.35,
x    70.76,   71.16,   73.06,   72.19,   71.64,   69.07,   67.12,
x    65.49,   63.44,   63.06,   61.58,   60.77,   65.60,   66.68,
x    66.29,   68.98,   71.42,   72.79,   77.06,   79.76,   78.60,
x    77.03,   76.87,   72.24,   69.79,   68.02,   66.03,   66.65,
x    66.04,   66.76,
x    53.20,   56.66,   60.73,   61.17,   63.06,   65.51,   69.65,
x    72.12,   71.51,   72.35,   71.84,   72.95,   71.24,   68.19,
x    66.75,   64.79,   64.72,   62.95,   62.07,   66.77,   68.37,
x    67.60,   69.63,   71.31,   76.12,   78.08,   77.63,   77.68,
x    77.53,   77.09,   74.25,   71.51,   71.45,   68.74,   69.44,
x    67.94,   66.81,
x    55.96,   58.00,   62.15,   62.54,   63.85,   67.91,   73.21,
x    73.49,   72.76,   73.55,   72.99,   73.09,   73.22,   69.24,
x    66.48,   64.69,   65.52,   63.80,   63.64,   67.54,   69.90,
x    69.24,   70.47,   75.04,   78.72,   78.77,   78.42,   77.64,
x    77.35,   77.01,   77.05,   72.73,   68.99,   67.28,   68.76,
x    68.17,   66.69,
x    59.36,   60.60,   64.76,   69.11,   71.30,   69.10,   70.12,
x    72.26,   72.32,   73.14,   72.90,   72.66,   70.59,   70.37,
x    68.91,   67.16,   66.49,   64.83,   64.80,   67.92,   70.90,
x    72.52,   75.64,   75.39,   76.05,   77.49,   77.40,   77.32,
x    76.51,   76.09,   74.39,   73.74,   74.75,   73.58,   72.23,
x    69.35,   68.69,
x    58.85,   59.73,   62.58,   65.30,   67.40,   67.54,   67.90,
x    69.67,   70.76,   71.95,   71.55,   70.13,   69.28,   68.49,
x    68.19,   67.22,   66.80,   65.67,   65.24,   66.54,   69.99,
x    72.86,   73.78,   73.51,   73.49,   74.94,   75.09,   75.61,
x    75.00,   73.41,   71.34,   70.61,   71.38,   70.54,   70.90,
x    68.86,   68.37,
x    58.28,   59.14,   61.15,   63.20,   65.71,   65.58,   65.92,
x    67.35,   68.14,   69.25,   69.15,   67.45,   66.65,   66.67,
x    66.90,   65.80,   66.09,   65.37,   65.00,   65.44,   67.47,
```

```
x    69.48,   71.18,   70.18,   70.19,   71.29,   72.23,   73.32,
x    72.47,   70.40,   68.46,   68.31,   68.49,   68.07,   68.28,
x    66.66,   66.70/

 data  ((doqt_cnx(i,j), i = 1,37),j=11,20)/
x    56.56,   57.36,   59.21,   59.91,   62.22,   63.64,   63.81,
x    64.57,   66.21,   65.64,   66.36,   64.36,   64.33,   65.14,
x    64.11,   62.64,   63.79,   63.06,   62.49,   62.92,   64.65,
x    65.18,   66.67,   68.38,   67.76,   68.28,   69.16,   68.61,
x    69.21,   67.11,   65.98,   65.78,   65.64,   64.87,   65.43,
x    63.88,   63.71,
x    53.34,   54.67,   53.35,   57.50,   58.31,   57.92,   59.93,
x    59.76,   62.10,   59.53,   61.78,   60.42,   61.38,   60.15,
x    61.10,   60.75,   58.22,   59.89,   59.32,   60.18,   58.62,
x    62.28,   63.27,   61.69,   63.81,   62.11,   64.68,   61.06,
x    63.35,   60.65,   61.20,   60.30,   61.44,   62.44,   59.23,
x    60.64,   60.45,
x    50.69,   50.12,   52.26,   54.66,   54.42,   56.20,   54.34,
x    55.76,   56.26,   55.13,   57.08,   57.13,   55.57,   57.70,
x    56.71,   57.50,   56.95,   53.48,   56.73,   54.49,   56.83,
x    58.49,   57.87,   59.42,   56.70,   58.51,   58.15,   56.25,
x    57.52,   57.26,   55.77,   58.25,   57.83,   58.59,   57.60,
x    53.84,   57.93,
x    47.60,   47.35,   49.51,   48.73,   52.26,   50.98,   51.43,
x    51.78,   51.84,   53.40,   51.26,   51.48,   51.48,   52.57,
x    53.85,   52.26,   52.99,   50.48,   53.18,   51.74,   53.13,
x    52.30,   54.92,   53.35,   52.51,   52.73,   51.83,   54.27,
x    51.92,   52.13,   52.06,   53.71,   54.97,   51.94,   53.22,
x    50.33,   54.30,
x    45.89,   45.73,   47.01,   46.69,   47.42,   49.17,   48.42,
x    46.61,   48.14,   48.57,   48.23,   46.64,   48.57,   50.01,
x    48.46,   48.44,   48.64,   46.60,   48.61,   46.93,   48.46,
x    48.12,   48.48,   50.54,   49.50,   47.60,   48.56,   48.75,
x    47.90,   46.96,   50.03,   51.04,   48.47,   49.40,   49.50,
x    47.25,   50.14,
x    44.21,   42.77,   43.35,   44.73,   44.17,   44.22,   45.30,
x    46.30,   44.31,   43.85,   43.90,   45.65,   45.83,   45.11,
x    45.07,   45.77,   43.97,   43.74,   45.19,   43.64,   43.53,
x    45.70,   44.78,   45.05,   46.24,   47.23,   45.36,   45.12,
x    45.44,   47.35,   45.05,   45.14,   46.16,   46.68,   44.62,
x    44.61,   46.54,
x    42.78,   43.07,   43.35,   42.68,   43.12,   43.36,   43.20,
x    43.27,   43.07,   43.41,   43.57,   43.41,   43.43,   43.56,
x    44.15,   44.03,   43.51,   43.31,   42.58,   43.57,   44.17,
x    44.09,   44.12,   44.02,   43.70,   43.79,   42.97,   43.83,
x    43.16,   44.11,   44.19,   44.72,   44.97,   44.92,   45.24,
x    44.80,   44.38,
x    43.29,   43.97,   43.92,   43.35,   43.04,   42.51,   42.94,
x    42.97,   42.93,   42.97,   42.87,   42.81,   42.66,   42.89,
x    42.79,   43.10,   43.28,   43.25,   42.28,   43.50,   42.84,
x    42.86,   42.80,   42.33,   42.37,   42.95,   42.48,   42.55,
x    42.48,   41.86,   41.30,   42.03,   41.87,   41.60,   41.77,
x    41.88,   42.84,
x    42.82,   42.82,   42.82,   42.82,   42.82,   42.82,   42.82,
x    42.82,   42.82,   42.82,   42.82,   42.82,   42.82,   42.82,
x    42.82,   42.82,   42.82,   42.82,   42.82,   42.82,   42.82,
x    42.82,   42.82,   42.82,   42.82,   42.82,   42.82,   42.82,
```

```
x    42.82,   42.82,   42.82,   42.82,   42.82,   42.82,   42.82,
x    42.82,   42.82,
x    42.84,   41.88,   41.77,   41.60,   41.87,   42.03,   41.30,
x    41.86,   42.48,   42.55,   42.48,   42.95,   42.37,   42.33,
x    42.80,   42.86,   42.84,   43.50,   42.28,   43.25,   43.28,
x    43.10,   42.79,   42.89,   42.66,   42.81,   42.87,   42.97,
x    42.93,   42.97,   42.94,   42.51,   43.04,   43.35,   43.92,
x    43.97,   43.29/

 data  ((doqt_cnx(i,j), i = 1,37),j=21,30)/
x    44.38,   44.80,   45.24,   44.92,   44.97,   44.72,   44.19,
x    44.11,   43.16,   43.83,   42.97,   43.79,   43.70,   44.02,
x    44.12,   44.09,   44.17,   43.57,   42.58,   43.31,   43.51,
x    44.03,   44.15,   43.56,   43.43,   43.41,   43.57,   43.41,
x    43.07,   43.27,   43.20,   43.36,   43.12,   42.68,   43.35,
x    43.07,   42.78,
x    46.54,   44.61,   44.62,   46.68,   46.16,   45.14,   45.05,
x    47.35,   45.44,   45.12,   45.36,   47.23,   46.24,   45.05,
x    44.78,   45.70,   43.53,   43.64,   45.19,   43.74,   43.97,
x    45.77,   45.07,   45.11,   45.83,   45.65,   43.90,   43.85,
x    44.31,   46.30,   45.30,   44.22,   44.17,   44.73,   43.35,
x    42.77,   44.21,
x    50.14,   47.25,   49.50,   49.40,   48.47,   51.04,   50.03,
x    46.96,   47.90,   48.75,   48.56,   47.60,   49.50,   50.54,
x    48.48,   48.12,   48.46,   46.93,   48.61,   46.60,   48.64,
x    48.44,   48.46,   50.01,   48.57,   46.64,   48.23,   48.57,
x    48.14,   46.61,   48.42,   49.17,   47.42,   46.69,   47.01,
x    45.73,   45.89,
x    54.30,   50.33,   53.22,   51.94,   54.97,   53.71,   52.06,
x    52.13,   51.92,   54.27,   51.83,   52.73,   52.51,   53.35,
x    54.92,   52.30,   53.13,   51.74,   53.18,   50.48,   52.99,
x    52.26,   53.85,   52.57,   51.48,   51.48,   51.26,   53.40,
x    51.84,   51.78,   51.43,   50.98,   52.26,   48.73,   49.51,
x    47.35,   47.60,
x    57.93,   53.84,   57.60,   58.59,   57.83,   58.25,   55.77,
x    57.26,   57.52,   56.25,   58.15,   58.51,   56.70,   59.42,
x    57.87,   58.49,   56.83,   54.49,   56.73,   53.48,   56.95,
x    57.50,   56.71,   57.70,   55.57,   57.13,   57.08,   55.13,
x    56.26,   55.76,   54.34,   56.20,   54.42,   54.66,   52.26,
x    50.12,   50.69,
x    60.45,   60.64,   59.23,   62.44,   61.44,   60.30,   61.20,
x    60.65,   63.35,   61.06,   64.68,   62.11,   63.81,   61.69,
x    63.27,   62.28,   58.62,   60.18,   59.32,   59.89,   58.22,
x    60.75,   61.10,   60.15,   61.38,   60.42,   61.78,   59.53,
x    62.10,   59.76,   59.93,   57.92,   58.31,   57.50,   53.35,
x    54.67,   53.34,
x    63.71,   63.88,   65.43,   64.87,   65.64,   65.78,   65.98,
x    67.11,   69.21,   68.61,   69.16,   68.28,   67.76,   68.38,
x    66.67,   65.18,   64.65,   62.92,   62.49,   63.06,   63.79,
x    62.64,   64.11,   65.14,   64.33,   64.36,   66.36,   65.64,
x    66.21,   64.57,   63.81,   63.64,   62.22,   59.91,   59.21,
x    57.36,   56.56,
x    66.70,   66.66,   68.28,   68.07,   68.49,   68.31,   68.46,
x    70.40,   72.47,   73.32,   72.23,   71.29,   70.19,   70.18,
x    71.18,   69.48,   67.47,   65.44,   65.00,   65.37,   66.09,
x    65.80,   66.90,   66.67,   66.65,   67.45,   69.15,   69.25,
x    68.14,   67.35,   65.92,   65.58,   65.71,   63.20,   61.15,
```

```
x    59.14,   58.28,
x    68.37,   68.86,   70.90,   70.54,   71.38,   70.61,   71.34,
x    73.41,   75.00,   75.61,   75.09,   74.94,   73.49,   73.51,
x    73.78,   72.86,   69.99,   66.54,   65.24,   65.67,   66.80,
x    67.22,   68.19,   68.49,   69.28,   70.13,   71.55,   71.95,
x    70.76,   69.67,   67.90,   67.54,   67.40,   65.30,   62.58,
x    59.73,   58.85,
x    68.69,   69.35,   72.23,   73.58,   74.75,   73.74,   74.39,
x    76.09,   76.51,   77.32,   77.40,   77.49,   76.05,   75.39,
x    75.64,   72.52,   70.90,   67.92,   64.80,   64.83,   66.49,
x    67.16,   68.91,   70.37,   70.59,   72.66,   72.90,   73.14,
x    72.32,   72.26,   70.12,   69.10,   71.30,   69.11,   64.76,
x    60.60,   59.36/

 data  ((doqt_cnx(i,j), i = 1,37),j=31,37)/
x    66.69,   68.17,   68.76,   67.28,   68.99,   72.73,   77.05,
x    77.01,   77.35,   77.64,   78.42,   78.77,   78.72,   75.04,
x    70.47,   69.24,   69.90,   67.54,   63.64,   63.80,   65.52,
x    64.69,   66.48,   69.24,   73.22,   73.09,   72.99,   73.55,
x    72.76,   73.49,   73.21,   67.91,   63.85,   62.54,   62.15,
x    58.00,   55.96,
x    66.81,   67.94,   69.44,   68.74,   71.45,   71.51,   74.25,
x    77.09,   77.53,   77.68,   77.63,   78.08,   76.12,   71.31,
x    69.63,   67.60,   68.37,   66.77,   62.07,   62.95,   64.72,
x    64.79,   66.75,   68.19,   71.24,   72.95,   71.84,   72.35,
x    71.51,   72.12,   69.65,   65.51,   63.06,   61.17,   60.73,
x    56.66,   53.20,
x    66.76,   66.04,   66.65,   66.03,   68.02,   69.79,   72.24,
x    76.87,   77.03,   78.60,   79.76,   77.06,   72.79,   71.42,
x    68.98,   66.29,   66.68,   65.60,   60.77,   61.58,   63.06,
x    63.44,   65.49,   67.12,   69.07,   71.64,   72.19,   73.06,
x    71.16,   70.76,   66.35,   64.17,   61.81,   59.55,   58.32,
x    54.56,   50.86,
x    65.93,   64.86,   65.78,   65.19,   66.92,   69.25,   73.11,
x    76.86,   79.65,   80.64,   81.44,   76.61,   74.03,   70.89,
x    67.75,   65.38,   65.46,   64.65,   59.72,   60.01,   61.48,
x    62.31,   64.43,   66.62,   69.57,   71.49,   73.17,   73.07,
x    72.24,   70.06,   66.67,   62.72,   59.71,   57.01,   55.47,
x    52.45,   47.74,
x    66.02,   65.17,   66.16,   65.89,   68.20,   70.78,   74.98,
x    79.58,   81.46,   82.03,   82.19,   78.81,   74.51,   71.05,
x    67.26,   64.97,   64.40,   64.07,   59.13,   59.39,   61.31,
x    62.78,   65.61,   68.10,   71.04,   73.27,   73.59,   72.89,
x    72.19,   70.31,   66.63,   63.17,   60.07,   56.82,   54.49,
x    50.93,   46.20,
x    66.87,   66.19,   66.39,   66.49,   69.27,   72.34,   76.47,
x    80.21,   81.48,   82.98,   83.05,   79.27,   75.83,   71.94,
x    68.28,   65.45,   64.50,   64.63,   61.13,   59.80,   61.35,
x    63.77,   67.42,   70.98,   73.64,   75.16,   74.92,   75.56,
x    73.65,   71.51,   68.39,   65.12,   60.68,   57.10,   54.36,
x    50.78,   45.09,
x    66.76,   65.99,   66.23,   65.78,   69.08,   70.91,   73.71,
x    76.56,   79.00,   80.24,   80.75,   78.85,   75.98,   72.61,
x    68.47,   66.15,   64.70,   64.43,   62.92,   59.20,   61.22,
x    64.59,   68.86,   72.16,   74.86,   76.02,   75.64,   76.02,
x    74.55,   72.92,   69.82,   66.58,   62.45,   57.85,   54.17,
x    50.83,   48.33/
```

```
 data  ((yoqt_cnx(i,j), i = 1,37),j=1,10)/
x     0.00,    0.00,    0.00,    0.00,    0.00,    0.00,    0.00,
x     0.00,    0.00,    0.00,    0.00,    0.00,    0.00,    0.00,
x     0.00,    0.00,    0.00,    0.00,    0.00,    0.00,    0.00,
x     0.00,    0.00,    0.00,    0.00,    0.00,    0.00,    0.00,
x     0.00,    0.00,    0.00,    0.00,    0.00,    0.00,    0.00,
x     0.00,    0.00,
x    10.82,   11.61,   11.18,    9.46,    6.53,    4.40,    1.76,
x     0.83,   -0.10,    0.49,    0.29,    1.66,    3.73,    6.37,
x     8.25,    9.74,   10.35,   11.68,   11.48,    9.74,   11.81,
x    10.16,    5.41,    2.05,    1.16,    0.69,    0.57,    0.72,
x     1.16,    4.36,    5.41,    6.69,    8.68,   10.15,   11.34,
x    10.21,    9.40,
x    11.04,   13.48,   13.26,   11.33,    8.64,    5.83,    3.01,
x     2.78,    3.05,    3.92,    3.87,    5.28,    6.40,    8.77,
x    11.65,   12.99,   14.96,   16.29,   18.03,   17.76,   16.61,
x    14.27,   10.36,    5.58,    1.77,    1.35,    0.59,    1.80,
x     2.39,    4.04,    5.63,    8.18,   11.80,   14.40,   17.53,
x    19.72,   20.05,
x     9.26,   11.80,   11.78,   10.74,    8.52,    6.57,    4.48,
x     4.55,    4.94,    5.51,    5.41,    6.79,    7.59,    9.68,
x    12.66,   14.03,   15.13,   16.27,   17.36,   17.67,   16.43,
x    13.99,   10.97,    8.77,    5.03,    4.14,    2.99,    3.74,
x     3.14,    4.52,    5.88,    8.75,   12.31,   15.21,   18.06,
x    20.01,   20.69,
x     7.80,    9.65,   10.49,    9.72,    9.15,    7.12,    7.24,
x     6.66,    7.06,    7.30,    7.74,    8.21,    9.92,   11.15,
x    13.37,   14.01,   14.78,   15.06,   16.00,   16.76,   15.79,
x    12.77,   10.90,   10.12,    8.64,    7.36,    7.20,    6.90,
x     6.59,    6.17,    8.61,   10.17,   12.21,   14.53,   16.79,
x    17.91,   18.36,
x     7.90,    8.93,   10.14,   10.15,    9.96,    9.98,    9.92,
x     8.51,    8.60,    9.52,    9.55,    9.42,   11.90,   12.76,
x    13.85,   14.17,   14.92,   14.72,   15.29,   16.52,   16.08,
x    13.33,   12.03,   11.80,   11.77,    9.56,    8.24,    9.54,
x     9.60,    9.05,   11.08,   12.84,   12.73,   13.90,   15.53,
x    15.76,   14.78,
x     7.88,    8.87,   10.45,   10.76,   10.56,   11.33,   12.31,
x    10.86,   11.19,   12.17,   11.43,   11.93,   13.74,   14.52,
x    14.77,   15.06,   15.45,   15.01,   15.72,   17.23,   17.76,
x    15.29,   13.65,   14.57,   14.72,   12.85,   11.76,   12.10,
x    11.56,   10.58,   12.83,   13.25,   12.84,   13.79,   14.99,
x    15.41,   15.87,
x     7.93,    7.91,    9.60,   12.65,   12.62,   12.57,   13.17,
x    14.31,   14.19,   14.80,   14.59,   15.24,   15.71,   16.09,
x    15.75,   15.40,   15.38,   15.37,   16.55,   17.28,   18.57,
x    17.32,   16.20,   16.17,   16.39,   16.16,   15.15,   15.45,
x    15.00,   14.84,   14.61,   14.12,   13.82,   14.36,   14.34,
x    14.36,   14.34,
x     9.36,    8.95,    8.89,   12.00,   13.01,   13.01,   13.74,
x    14.47,   15.53,   16.88,   16.29,   16.06,   16.32,   16.35,
x    16.24,   15.58,   15.66,   15.86,   17.21,   16.58,   16.29,
x    18.01,   17.52,   17.00,   16.76,   16.79,   17.15,   17.94,
x    16.73,   15.94,   15.27,   14.16,   13.75,   14.48,   15.56,
x    16.07,   17.01,
x     9.13,    8.37,    7.86,   10.97,   13.39,   12.76,   13.16,
```

```
     x    14.15,   14.87,   15.96,   15.98,   15.93,   16.06,   16.55,
     x    16.65,   15.59,   16.15,   17.38,   16.97,   14.80,   13.94,
     x    15.78,   17.10,   16.18,   16.32,   16.68,   16.75,   17.09,
     x    16.66,   16.02,   14.73,   14.14,   13.82,   14.57,   15.12,
     x    15.51,   16.43/

      data  ((yoqt_cnx(i,j), i = 1,37),j=11,20)/
     x     8.95,    8.00,    7.05,    7.99,   10.66,   12.34,   12.79,
     x    13.65,   14.43,   15.04,   15.21,   15.31,   15.61,   16.54,
     x    15.78,   14.41,   15.90,   13.74,   13.61,   12.88,   11.29,
     x    11.08,   12.70,   12.95,   14.00,   15.16,   15.90,   15.74,
     x    15.69,   14.62,   14.07,   13.80,   14.05,   13.96,   13.93,
     x    14.36,   15.29,
     x     7.71,    6.99,    5.89,    3.47,    8.33,    5.99,   10.30,
     x    10.08,   13.57,   11.83,   13.88,   13.09,   14.43,   14.24,
     x    14.42,   13.22,    8.24,    7.03,    5.35,    6.05,    4.96,
     x     4.61,    5.89,    4.40,    8.42,   10.50,   12.76,   12.47,
     x    13.45,   12.29,   12.73,   11.98,   12.63,   11.54,    8.70,
     x    10.06,    9.81,
     x     4.99,    4.71,    4.54,    2.54,    0.79,    1.84,    6.52,
     x     8.45,    9.55,    9.48,   10.88,   11.09,   10.14,   12.51,
     x    11.86,   11.22,    4.50,    6.22,    1.82,    2.33,    2.21,
     x     2.75,    1.70,    4.43,    9.21,   10.70,   10.39,    9.66,
     x    10.35,    9.99,    8.85,   10.91,   10.82,    9.93,    5.02,
     x     7.00,    3.46,
     x     2.91,    2.31,    1.45,    2.68,   -0.24,    1.74,    7.37,
     x     7.46,    7.00,    8.40,    6.81,    6.57,    7.12,    9.68,
     x    10.81,    8.20,    3.90,    6.25,    0.97,    0.68,    1.07,
     x     6.05,    2.87,    5.84,    7.42,    6.90,    6.21,    8.34,
     x     6.46,    6.16,    6.61,    7.81,    8.78,    8.25,    6.23,
     x     6.29,    2.35,
     x    -1.67,    0.60,    1.63,    0.75,    3.64,    3.64,    2.06,
     x     1.45,    4.33,    4.83,    4.99,    4.33,    6.27,    7.58,
     x     5.96,    3.00,    3.51,    3.30,    0.22,    2.99,    2.25,
     x     1.99,    5.40,    5.14,    4.00,    2.33,    3.59,    3.93,
     x     3.99,    2.37,    3.90,    5.74,    5.84,    3.82,    2.51,
     x     3.80,    1.20,
     x    -6.08,   -2.40,   -1.82,   -1.72,   -3.72,   -1.58,    2.11,
     x    -1.19,   -3.13,   -3.45,   -1.43,    3.59,    2.96,   -0.24,
     x    -1.57,    1.33,    0.34,    0.59,   -2.84,    0.14,    0.03,
     x     0.73,   -1.27,   -0.19,    2.70,    0.61,   -3.22,   -3.96,
     x    -3.24,    0.23,    1.76,   -0.42,   -3.31,    0.23,   -2.33,
     x    -2.03,   -1.55,
     x    -8.30,   -5.22,   -5.66,   -4.83,   -5.02,   -6.98,   -7.36,
     x    -7.00,   -9.73,   -6.99,  -11.92,   -8.71,   -8.04,   -6.78,
     x    -7.65,   -7.82,   -7.24,   -6.32,   -6.18,   -6.99,   -7.98,
     x    -8.26,   -7.89,   -6.97,   -7.87,   -8.40,  -10.39,   -6.64,
     x    -9.66,   -7.81,   -7.50,   -7.02,   -7.63,   -7.72,   -7.35,
     x    -6.70,   -5.77,
     x    -6.72,  -11.30,  -10.36,  -10.39,  -11.16,  -10.16,  -11.99,
     x   -11.34,  -11.65,  -10.53,  -10.34,   -8.61,   -8.50,   -7.60,
     x    -6.96,   -6.90,   -7.13,   -7.11,   -6.19,   -6.32,   -7.16,
     x    -7.78,   -8.16,   -7.88,   -8.91,   -9.87,   -9.35,   -9.33,
     x    -9.77,   -8.91,   -8.08,   -8.51,   -8.09,   -7.68,   -7.54,
     x    -7.44,   -6.49,
     x    -1.95,   -1.93,   -1.41,   -1.36,   -1.53,   -0.83,   -1.96,
     x    -1.22,   -0.94,   -0.60,   -0.50,    0.63,    0.21,    0.14,
```

```
x      0.60,     0.44,     0.01,    -0.39,     0.00,     0.39,    -0.01,
x     -0.44,    -0.60,    -0.14,    -0.21,    -0.63,     0.50,     0.60,
x      0.94,     1.22,     1.96,     0.83,     1.53,     1.36,     1.41,
x      1.93,     1.95,
x      6.49,     7.44,     7.54,     7.68,     8.09,     8.51,     8.08,
x      8.91,     9.77,     9.33,     9.35,     9.87,     8.91,     7.88,
x      8.16,     7.78,     7.16,     6.32,     6.19,     7.11,     7.13,
x      6.90,     6.96,     7.60,     8.50,     8.61,    10.34,    10.53,
x     11.65,    11.34,    11.99,    10.16,    11.16,    10.39,    10.36,
x     11.30,     6.72/

  data  ((yoqt_cnx(i,j), i = 1,37),j=21,30)/
x      5.77,     6.70,     7.35,     7.72,     7.63,     7.02,     7.50,
x      7.81,     9.66,     6.64,    10.39,     8.40,     7.87,     6.97,
x      7.89,     8.26,     7.98,     6.99,     6.18,     6.32,     7.24,
x      7.82,     7.65,     6.78,     8.04,     8.71,    11.92,     6.99,
x      9.73,     7.00,     7.36,     6.98,     5.02,     4.83,     5.66,
x      5.22,     8.30,
x      1.55,     2.03,     2.33,    -0.23,     3.31,     0.42,    -1.76,
x     -0.23,     3.24,     3.96,     3.22,    -0.61,    -2.70,     0.19,
x      1.27,    -0.73,    -0.03,    -0.14,     2.84,    -0.59,    -0.34,
x     -1.33,     1.57,     0.24,    -2.96,    -3.59,     1.43,     3.45,
x      3.13,     1.19,    -2.11,     1.58,     3.72,     1.72,     1.82,
x      2.40,     6.08,
x     -1.20,    -3.80,    -2.51,    -3.82,    -5.84,    -5.74,    -3.90,
x     -2.37,    -3.99,    -3.93,    -3.59,    -2.33,    -4.00,    -5.14,
x     -5.40,    -1.99,    -2.25,    -2.99,    -0.22,    -3.30,    -3.51,
x     -3.00,    -5.96,    -7.58,    -6.27,    -4.33,    -4.99,    -4.83,
x     -4.33,    -1.45,    -2.06,    -3.64,    -3.64,    -0.75,    -1.63,
x     -0.60,     1.67,
x     -2.35,    -6.29,    -6.23,    -8.25,    -8.78,    -7.81,    -6.61,
x     -6.16,    -6.46,    -8.34,    -6.21,    -6.90,    -7.42,    -5.84,
x     -2.87,    -6.05,    -1.07,    -0.68,    -0.97,    -6.25,    -3.90,
x     -8.20,   -10.81,    -9.68,    -7.12,    -6.57,    -6.81,    -8.40,
x     -7.00,    -7.46,    -7.37,    -1.74,     0.24,    -2.68,    -1.45,
x     -2.31,    -2.91,
x     -3.46,    -7.00,    -5.02,    -9.93,   -10.82,   -10.91,    -8.85,
x     -9.99,   -10.35,    -9.66,   -10.39,   -10.70,    -9.21,    -4.43,
x     -1.70,    -2.75,    -2.21,    -2.33,    -1.82,    -6.22,    -4.50,
x    -11.22,   -11.86,   -12.51,   -10.14,   -11.09,   -10.88,    -9.48,
x     -9.55,    -8.45,    -6.52,    -1.84,    -0.79,    -2.54,    -4.54,
x     -4.71,    -4.99,
x     -9.81,   -10.06,    -8.70,   -11.54,   -12.63,   -11.98,   -12.73,
x    -12.29,   -13.45,   -12.47,   -12.76,   -10.50,    -8.42,    -4.40,
x     -5.89,    -4.61,    -4.96,    -6.05,    -5.35,    -7.03,    -8.24,
x    -13.22,   -14.42,   -14.24,   -14.43,   -13.09,   -13.88,   -11.83,
x    -13.57,   -10.08,   -10.30,    -5.99,    -8.33,    -3.47,    -5.89,
x     -6.99,    -7.71,
x    -15.29,   -14.36,   -13.93,   -13.96,   -14.05,   -13.80,   -14.07,
x    -14.62,   -15.69,   -15.74,   -15.90,   -15.16,   -14.00,   -12.95,
x    -12.70,   -11.08,   -11.29,   -12.88,   -13.61,   -13.74,   -15.90,
x    -14.41,   -15.78,   -16.54,   -15.61,   -15.31,   -15.21,   -15.04,
x    -14.43,   -13.65,   -12.79,   -12.34,   -10.66,    -7.99,    -7.05,
x     -8.00,    -8.95,
x    -16.43,   -15.51,   -15.12,   -14.57,   -13.82,   -14.14,   -14.73,
x    -16.02,   -16.66,   -17.09,   -16.75,   -16.68,   -16.32,   -16.18,
x    -17.10,   -15.78,   -13.94,   -14.80,   -16.97,   -17.38,   -16.15,
```

```
x   -15.59,  -16.65,  -16.55,  -16.06,  -15.93,  -15.98,  -15.96,
x   -14.87,  -14.15,  -13.16,  -12.76,  -13.39,  -10.97,   -7.86,
x    -8.37,   -9.13,
x   -17.01,  -16.07,  -15.56,  -14.48,  -13.75,  -14.16,  -15.27,
x   -15.94,  -16.73,  -17.94,  -17.15,  -16.79,  -16.76,  -17.00,
x   -17.52,  -18.01,  -16.29,  -16.58,  -17.21,  -15.86,  -15.66,
x   -15.58,  -16.24,  -16.35,  -16.32,  -16.06,  -16.29,  -16.88,
x   -15.53,  -14.47,  -13.74,  -13.01,  -13.01,  -12.00,   -8.89,
x    -8.95,   -9.36,
x   -14.34,  -14.36,  -14.34,  -14.36,  -13.82,  -14.12,  -14.61,
x   -14.84,  -15.00,  -15.45,  -15.15,  -16.16,  -16.39,  -16.17,
x   -16.20,  -17.32,  -18.57,  -17.28,  -16.55,  -15.37,  -15.38,
x   -15.40,  -15.75,  -16.09,  -15.71,  -15.24,  -14.59,  -14.80,
x   -14.19,  -14.31,  -13.17,  -12.57,  -12.62,  -12.65,   -9.60,
x    -7.91,   -7.93/

 data  ((yoqt_cnx(i,j), i = 1,37),j=31,37)/
x   -15.87,  -15.41,  -14.99,  -13.79,  -12.84,  -13.25,  -12.83,
x   -10.58,  -11.56,  -12.10,  -11.76,  -12.85,  -14.72,  -14.57,
x   -13.65,  -15.29,  -17.76,  -17.23,  -15.72,  -15.01,  -15.45,
x   -15.06,  -14.77,  -14.52,  -13.74,  -11.93,  -11.43,  -12.17,
x   -11.19,  -10.86,  -12.31,  -11.33,  -10.56,  -10.76,  -10.45,
x    -8.87,   -7.88,
x   -14.78,  -15.76,  -15.53,  -13.90,  -12.73,  -12.84,  -11.08,
x    -9.05,   -9.60,   -9.54,   -8.24,   -9.56,  -11.77,  -11.80,
x   -12.03,  -13.33,  -16.08,  -16.52,  -15.29,  -14.72,  -14.92,
x   -14.17,  -13.85,  -12.76,  -11.90,   -9.42,   -9.55,   -9.52,
x    -8.60,   -8.51,   -9.92,   -9.98,   -9.96,  -10.15,  -10.14,
x    -8.93,   -7.90,
x   -18.36,  -17.91,  -16.79,  -14.53,  -12.21,  -10.17,   -8.61,
x    -6.17,   -6.59,   -6.90,   -7.20,   -7.36,   -8.64,  -10.12,
x   -10.90,  -12.77,  -15.79,  -16.76,  -16.00,  -15.06,  -14.78,
x   -14.01,  -13.37,  -11.15,   -9.92,   -8.21,   -7.74,   -7.30,
x    -7.06,   -6.66,   -7.24,   -7.12,   -9.15,   -9.72,  -10.49,
x    -9.65,   -7.80,
x   -20.69,  -20.01,  -18.06,  -15.21,  -12.31,   -8.75,   -5.88,
x    -4.52,   -3.14,   -3.74,   -2.99,   -4.14,   -5.03,   -8.77,
x   -10.97,  -13.99,  -16.43,  -17.67,  -17.36,  -16.27,  -15.13,
x   -14.03,  -12.66,   -9.68,   -7.59,   -6.79,   -5.41,   -5.51,
x    -4.94,   -4.55,   -4.48,   -6.57,   -8.52,  -10.74,  -11.78,
x   -11.80,   -9.26,
x   -20.05,  -19.72,  -17.53,  -14.40,  -11.80,   -8.18,   -5.63,
x    -4.04,   -2.39,   -1.80,   -0.59,   -1.35,   -1.77,   -5.58,
x   -10.36,  -14.27,  -16.61,  -17.76,  -18.03,  -16.29,  -14.96,
x   -12.99,  -11.65,   -8.77,   -6.40,   -5.28,   -3.87,   -3.92,
x    -3.05,   -2.78,   -3.01,   -5.83,   -8.64,  -11.33,  -13.26,
x   -13.48,  -11.04,
x    -9.40,  -10.21,  -11.34,  -10.15,   -8.68,   -6.69,   -5.41,
x    -4.36,   -1.16,   -0.72,   -0.57,   -0.69,   -1.16,   -2.05,
x    -5.41,  -10.16,  -11.81,   -9.74,  -11.48,  -11.68,  -10.35,
x    -9.74,   -8.25,   -6.37,   -3.73,   -1.66,   -0.29,   -0.49,
x     0.10,   -0.83,   -1.76,   -4.40,   -6.53,   -9.46,  -11.18,
x   -11.61,  -10.82,
x    -3.06,   -1.97,   -3.08,   -1.27,   -2.27,   -2.17,   -2.04,
x    -1.11,   -0.01,   -0.06,    1.10,    0.57,    0.24,   -0.50,
x    -0.35,   -0.07,   -1.05,   -0.51,   -0.73,   -1.07,   -0.79,
x    -1.12,   -1.07,   -0.37,   -0.48,   -0.29,    0.95,    0.95,
```

```
x     0.86,    0.83,    0.37,    0.23,   -0.21,   -0.61,   -0.26,
x    -0.92,    0.48/

 data  ((loqt_cnx(i,j), i = 1,37),j=1,10)/
x     0.00,  -13.37,  -13.39,  -10.36,   -8.78,   -6.29,   -3.12,
x    -2.19,    3.58,    3.67,    6.26,   10.08,   10.85,   11.99,
x    13.22,   14.79,   16.89,   17.13,    0.82,   -9.52,  -17.88,
x   -18.90,  -11.76,   -5.54,   -1.19,    1.00,   -1.58,    2.64,
x     4.10,   10.91,   11.57,   12.62,   14.12,   16.08,   20.70,
x    12.17,    0.00,
x     0.00,  -11.26,  -12.52,   -9.72,   -7.00,   -5.18,   -1.99,
x    -0.73,    4.23,    4.59,    5.45,   10.14,   10.88,   11.94,
x    12.17,   13.98,   16.08,   16.46,    7.74,   -9.46,  -16.98,
x   -17.88,  -11.96,   -5.37,   -1.00,    1.36,    4.73,    5.09,
x     4.86,    9.50,    9.86,   10.69,   13.29,   16.33,   19.45,
x    13.87,    0.00,
x     0.00,   -9.05,  -10.87,   -8.44,   -5.88,   -3.42,   -0.30,
x     2.00,    3.75,    5.19,    5.88,    7.22,    8.63,    9.27,
x    10.61,   12.38,   14.11,   14.56,   12.28,   -7.92,  -15.51,
x   -15.45,  -10.26,   -4.68,   -0.02,    2.73,    4.20,    6.72,
x     5.94,    6.87,    7.06,    8.80,   11.11,   14.11,   17.59,
x    15.40,    0.00,
x     0.00,   -6.26,   -8.32,   -5.48,   -3.52,   -0.75,    1.40,
x     2.65,    3.94,    4.60,    5.04,    4.98,    6.75,    8.48,
x     9.21,   11.10,   12.47,   13.30,   11.02,   -8.39,  -12.90,
x   -11.55,   -6.65,   -2.19,    2.42,    4.19,    5.63,    4.55,
x     2.75,    2.86,    3.47,    6.09,    9.09,   12.22,   14.46,
x    13.17,    0.00,
x     0.00,   -1.63,   -4.73,   -2.88,   -1.59,   -0.06,    2.12,
x     3.53,    4.24,    4.41,    4.76,    4.41,    5.25,    6.83,
x     6.38,    8.03,    9.82,   10.70,    9.07,   -5.22,   -9.75,
x    -7.30,   -3.21,   -0.87,    2.69,    5.28,    5.33,    3.53,
x     2.82,    1.49,    3.28,    4.24,    6.49,    9.44,   11.36,
x     9.26,    0.00,
x     0.00,    1.86,   -2.12,   -0.83,    0.32,    2.06,    3.94,
x     4.86,    4.21,    4.50,    4.46,    2.51,    2.77,    4.35,
x     4.31,    5.56,    7.00,    7.52,    5.60,   -2.92,   -5.58,
x    -4.34,   -0.51,    1.38,    4.30,    5.54,    4.37,    2.67,
x     2.58,    1.66,    1.17,    1.70,    2.66,    4.06,    7.72,
x     6.64,    0.00,
x     0.00,    4.61,    0.89,    0.97,    1.53,    4.13,    5.61,
x     5.67,    4.96,    4.21,    3.54,    1.90,    1.12,    1.95,
x     2.53,    3.96,    4.73,    4.89,    3.78,   -0.26,   -1.22,
x    -1.84,    1.26,    4.87,    6.30,    6.02,    4.65,    2.77,
x     2.33,    0.04,   -0.93,    0.44,    2.88,    5.37,    5.81,
x     4.09,    0.00,
x     0.00,    7.12,    5.09,   11.53,   11.78,    6.83,    5.40,
x     4.49,    3.85,    3.38,    3.21,    1.85,    1.13,    0.32,
x     0.17,    1.52,    3.40,    3.41,    4.00,    2.50,    2.61,
x     6.71,    9.49,    7.37,    6.21,    5.22,    4.01,    2.53,
x     2.10,    0.02,   -0.41,   -1.05,   -2.44,   -2.50,    0.30,
x     0.90,    0.00,
x     0.00,    5.96,   10.14,   10.77,   10.05,    8.33,    5.64,
x     4.93,    4.34,    3.55,    2.95,    1.57,    0.01,   -0.63,
x    -1.97,   -2.78,   -4.89,   -0.18,    2.89,    7.87,   11.20,
x    12.28,   11.11,    8.35,    6.19,    5.29,    3.85,    2.32,
x     1.74,    0.49,   -0.55,   -1.13,   -1.23,   -1.71,   -1.38,
```

```
x    -0.94,    0.00,
x     0.00,    5.67,   10.64,   12.35,   11.11,    9.06,    6.47,
x     5.74,    5.04,    3.62,    2.31,    0.27,   -0.21,   -1.12,
x    -2.50,   -3.33,   -6.92,   -1.27,    3.05,    8.68,   12.99,
x    13.00,   10.65,    7.99,    6.33,    5.11,    4.20,    2.84,
x     1.49,   -0.62,   -1.16,   -1.77,   -1.66,   -2.53,   -2.47,
x    -2.56,    0.00/

 data  ((loqt_cnx(i,j), i = 1,37),j=11,20)/
x     0.00,    5.27,    9.45,   10.67,   11.56,    9.44,    6.90,
x     6.39,    5.57,    3.61,    1.56,    0.10,   -0.54,   -2.09,
x    -2.57,   -4.39,   -7.00,   -1.66,    2.08,    7.15,   10.47,
x     9.82,   11.02,   11.65,    8.18,    7.34,    4.59,    3.15,
x     0.96,   -1.13,   -1.46,   -2.04,   -2.97,   -2.54,   -2.49,
x    -2.97,    0.00,
x     0.00,    5.73,    6.79,    7.81,    9.29,    7.92,    9.06,
x     8.84,    6.54,    4.33,    0.45,   -1.09,   -1.92,   -2.15,
x    -2.44,   -8.14,   -3.12,   -2.12,   -1.20,    3.39,    2.80,
x     6.15,    9.89,    8.86,   11.07,    7.83,    8.16,    3.63,
x    -0.26,   -1.30,   -1.72,   -2.50,   -2.20,   -2.04,   -1.04,
x    -2.65,    0.00,
x     0.00,    3.07,    5.87,    6.20,    5.69,    6.91,    7.77,
x     6.84,    5.94,    4.11,    1.53,   -0.75,   -2.23,   -2.33,
x    -5.67,   -6.91,   -0.96,   -3.15,   -3.63,   -1.78,   -0.42,
x     3.15,    3.20,    6.75,    5.65,    5.42,    4.88,    3.75,
x     0.25,   -1.57,   -2.85,   -2.44,   -1.75,   -1.19,   -0.24,
x    -0.78,    0.00,
x     0.00,    2.71,    5.20,    5.68,    4.67,    6.11,    7.56,
x     7.77,    6.41,    4.39,    1.22,   -1.34,   -1.83,   -5.88,
x    -5.48,   -2.75,   -0.48,   -2.60,   -3.00,   -2.38,   -0.33,
x     2.48,    2.64,    4.18,    5.69,    5.36,    4.20,    3.99,
x     1.01,   -1.81,   -2.93,   -3.08,   -2.76,   -1.53,   -1.37,
x    -0.85,    0.00,
x     0.00,    2.18,    4.63,    5.66,    5.20,    6.69,    7.39,
x     7.38,    6.40,    4.59,    2.56,   -2.12,   -5.91,   -5.36,
x    -3.57,   -1.30,   -0.86,   -2.37,   -1.62,    0.69,    1.66,
x     2.29,    4.15,    4.90,    5.80,    5.30,    3.75,    2.07,
x     0.48,   -1.49,   -2.91,   -3.21,   -3.17,   -1.80,   -0.55,
x    -0.82,    0.00,
x     0.00,    1.63,    3.56,    5.21,    4.50,    5.21,    6.96,
x     6.98,    2.40,    0.99,    0.39,   -3.63,   -2.75,   -1.88,
x    -0.57,   -2.08,   -2.44,   -1.84,   -1.05,    0.30,    1.60,
x     3.21,    3.22,    4.09,    5.47,    5.39,    3.14,    1.95,
x     1.09,   -0.68,   -3.58,   -1.82,    0.77,   -1.13,    0.43,
x     0.65,    0.00,
x     0.00,   -0.36,   -0.70,   -0.37,   -0.76,    0.17,   -0.17,
x    -0.16,    0.84,    1.66,    3.51,    2.75,    2.62,    1.98,
x     1.06,    0.28,   -0.98,   -0.42,   -1.09,   -1.01,   -1.04,
x    -1.00,   -0.19,    0.50,    0.21,    0.62,   -0.25,    1.18,
x     1.31,    1.78,    2.06,    2.22,    3.31,    3.78,    3.69,
x     1.65,    0.00,
x     0.00,   -0.04,    0.28,    0.52,    0.93,    0.98,    2.01,
x     1.92,    2.30,    2.33,    2.30,    1.43,    1.68,    1.46,
x     1.23,    1.00,    0.96,    0.88,   -0.90,   -0.55,   -0.70,
x    -1.40,   -1.95,   -1.44,   -0.94,   -0.78,   -0.02,    0.41,
x     1.09,    0.86,    1.00,    1.14,    1.34,    0.66,    0.88,
x     0.86,    0.00,
```

```
x     0.00,    0.41,    0.58,    0.59,    1.13,    1.06,    1.51,
x     1.39,    1.70,    1.37,    1.13,    0.32,    0.37,    0.01,
x    -0.36,   -0.19,    0.13,    0.17,   -0.36,    0.17,    0.13,
x    -0.19,   -0.36,    0.01,    0.37,    0.32,    1.13,    1.37,
x     1.70,    1.39,    1.51,    1.06,    1.13,    0.59,    0.58,
x     0.41,    0.00,
x     0.00,    0.86,    0.88,    0.66,    1.34,    1.14,    1.00,
x     0.86,    1.09,    0.41,   -0.02,   -0.78,   -0.94,   -1.44,
x    -1.95,   -1.40,   -0.70,   -0.55,   -0.90,    0.88,    0.96,
x     1.00,    1.23,    1.46,    1.68,    1.43,    2.30,    2.33,
x     2.30,    1.92,    2.01,    0.98,    0.93,    0.52,    0.28,
x    -0.04,    0.00/

 data  ((loqt_cnx(i,j), i = 1,37),j=21,30)/
x     0.00,    1.65,    3.69,    3.78,    3.31,    2.22,    2.06,
x     1.78,    1.31,    1.18,   -0.25,    0.62,    0.21,    0.50,
x    -0.19,   -1.00,   -1.04,   -1.01,   -1.09,   -0.42,   -0.98,
x     0.28,    1.06,    1.98,    2.62,    2.75,    3.51,    1.66,
x     0.84,   -0.16,   -0.17,    0.17,   -0.76,   -0.37,   -0.70,
x    -0.36,    0.00,
x     0.00,    0.65,    0.43,   -1.13,    0.77,   -1.82,   -3.58,
x    -0.68,    1.09,    1.95,    3.14,    5.39,    5.47,    4.09,
x     3.22,    3.21,    1.60,    0.30,   -1.05,   -1.84,   -2.44,
x    -2.08,   -0.57,   -1.88,   -2.75,   -3.63,    0.39,    0.99,
x     2.40,    6.98,    6.96,    5.21,    4.50,    5.21,    3.56,
x     1.63,    0.00,
x     0.00,   -0.82,   -0.55,   -1.80,   -3.17,   -3.21,   -2.91,
x    -1.49,    0.48,    2.07,    3.75,    5.30,    5.80,    4.90,
x     4.15,    2.29,    1.66,    0.69,   -1.62,   -2.37,   -0.86,
x    -1.30,   -3.57,   -5.36,   -5.91,   -2.12,    2.56,    4.59,
x     6.40,    7.38,    7.39,    6.69,    5.20,    5.66,    4.63,
x     2.18,    0.00,
x     0.00,   -0.85,   -1.37,   -1.53,   -2.76,   -3.08,   -2.93,
x    -1.81,    1.01,    3.99,    4.20,    5.36,    5.69,    4.18,
x     2.64,    2.48,   -0.33,   -2.38,   -3.00,   -2.60,   -0.48,
x    -2.75,   -5.48,   -5.88,   -1.83,   -1.34,    1.22,    4.39,
x     6.41,    7.77,    7.56,    6.11,    4.67,    5.68,    5.20,
x     2.71,    0.00,
x     0.00,   -0.78,   -0.24,   -1.19,   -1.75,   -2.44,   -2.85,
x    -1.57,    0.25,    3.75,    4.88,    5.42,    5.65,    6.75,
x     3.20,    3.15,   -0.42,   -1.78,   -3.63,   -3.15,   -0.96,
x    -6.91,   -5.67,   -2.33,   -2.23,   -0.75,    1.53,    4.11,
x     5.94,    6.84,    7.77,    6.91,    5.69,    6.20,    5.87,
x     3.07,    0.00,
x     0.00,   -2.65,   -1.04,   -2.04,   -2.20,   -2.50,   -1.72,
x    -1.30,   -0.26,    3.63,    8.16,    7.83,   11.07,    8.86,
x     9.89,    6.15,    2.80,    3.39,   -1.20,   -2.12,   -3.12,
x    -8.14,   -2.44,   -2.15,   -1.92,   -1.09,    0.45,    4.33,
x     6.54,    8.84,    9.06,    7.92,    9.29,    7.81,    6.79,
x     5.73,    0.00,
x     0.00,   -2.97,   -2.49,   -2.54,   -2.97,   -2.04,   -1.46,
x    -1.13,    0.96,    3.15,    4.59,    7.34,    8.18,   11.65,
x    11.02,    9.82,   10.47,    7.15,    2.08,   -1.66,   -7.00,
x    -4.39,   -2.57,   -2.09,   -0.54,    0.10,    1.56,    3.61,
x     5.57,    6.39,    6.90,    9.44,   11.56,   10.67,    9.45,
x     5.27,    0.00,
x     0.00,   -2.56,   -2.47,   -2.53,   -1.66,   -1.77,   -1.16,
```

```
x    -0.62,    1.49,    2.84,    4.20,    5.11,    6.33,    7.99,
x    10.65,   13.00,   12.99,    8.68,    3.05,   -1.27,   -6.92,
x    -3.33,   -2.50,   -1.12,   -0.21,    0.27,    2.31,    3.62,
x     5.04,    5.74,    6.47,    9.06,   11.11,   12.35,   10.64,
x     5.67,    0.00,
x     0.00,   -0.94,   -1.38,   -1.71,   -1.23,   -1.13,   -0.55,
x     0.49,    1.74,    2.32,    3.85,    5.29,    6.19,    8.35,
x    11.11,   12.28,   11.20,    7.87,    2.89,   -0.18,   -4.89,
x    -2.78,   -1.97,   -0.63,    0.01,    1.57,    2.95,    3.55,
x     4.34,    4.93,    5.64,    8.33,   10.05,   10.77,   10.14,
x     5.96,    0.00,
x     0.00,    0.90,    0.30,   -2.50,   -2.44,   -1.05,   -0.41,
x     0.02,    2.10,    2.53,    4.01,    5.22,    6.21,    7.37,
x     9.49,    6.71,    2.61,    2.50,    4.00,    3.41,    3.40,
x     1.52,    0.17,    0.32,    1.13,    1.85,    3.21,    3.38,
x     3.85,    4.49,    5.40,    6.83,   11.78,   11.53,    5.09,
x     7.12,    0.00/

 data  ((loqt_cnx(i,j), i = 1,37),j=31,37)/
x     0.00,    4.09,    5.81,    5.37,    2.88,    0.44,   -0.93,
x     0.04,    2.33,    2.77,    4.65,    6.02,    6.30,    4.87,
x     1.26,   -1.84,   -1.22,   -0.26,    3.78,    4.89,    4.73,
x     3.96,    2.53,    1.95,    1.12,    1.90,    3.54,    4.21,
x     4.96,    5.67,    5.61,    4.13,    1.53,    0.97,    0.89,
x     4.61,    0.00,
x     0.00,    6.64,    7.72,    4.06,    2.66,    1.70,    1.17,
x     1.66,    2.58,    2.67,    4.37,    5.54,    4.30,    1.38,
x    -0.51,   -4.34,   -5.58,   -2.92,    5.60,    7.52,    7.00,
x     5.56,    4.31,    4.35,    2.77,    2.51,    4.46,    4.50,
x     4.21,    4.86,    3.94,    2.06,    0.32,   -0.83,   -2.12,
x     1.86,    0.00,
x     0.00,    9.26,   11.36,    9.44,    6.49,    4.24,    3.28,
x     1.49,    2.82,    3.53,    5.33,    5.28,    2.69,   -0.87,
x    -3.21,   -7.30,   -9.75,   -5.22,    9.07,   10.70,    9.82,
x     8.03,    6.38,    6.83,    5.25,    4.41,    4.76,    4.41,
x     4.24,    3.53,    2.12,   -0.06,   -1.59,   -2.88,   -4.73,
x    -1.63,    0.00,
x     0.00,   13.17,   14.46,   12.22,    9.09,    6.09,    3.47,
x     2.86,    2.75,    4.55,    5.63,    4.19,    2.42,   -2.19,
x    -6.65,  -11.55,  -12.90,   -8.39,   11.02,   13.30,   12.47,
x    11.10,    9.21,    8.48,    6.75,    4.98,    5.04,    4.60,
x     3.94,    2.65,    1.40,   -0.75,   -3.52,   -5.48,   -8.32,
x    -6.26,    0.00,
x     0.00,   15.40,   17.59,   14.11,   11.11,    8.80,    7.06,
x     6.87,    5.94,    6.72,    4.20,    2.73,   -0.02,   -4.68,
x   -10.26,  -15.45,  -15.51,   -7.92,   12.28,   14.56,   14.11,
x    12.38,   10.61,    9.27,    8.63,    7.22,    5.88,    5.19,
x     3.75,    2.00,   -0.30,   -3.42,   -5.88,   -8.44,  -10.87,
x    -9.05,    0.00,
x     0.00,   13.87,   19.45,   16.33,   13.29,   10.69,    9.86,
x     9.50,    4.86,    5.09,    4.73,    1.36,   -1.00,   -5.37,
x   -11.96,  -17.88,  -16.98,   -9.46,    7.74,   16.46,   16.08,
x    13.98,   12.17,   11.94,   10.88,   10.14,    5.45,    4.59,
x     4.23,   -0.73,   -1.99,   -5.18,   -7.00,   -9.72,  -12.52,
x   -11.26,    0.00,
x     0.00,   12.17,   20.70,   16.08,   14.12,   12.62,   11.57,
x    10.91,    4.10,    2.64,   -1.58,    1.00,   -1.19,   -5.54,
```

```
x   -11.76,  -18.90,  -17.88,   -9.52,    0.82,   17.13,   16.89,
x    14.79,   13.22,   11.99,   10.85,   10.08,    6.26,    3.67,
x     3.58,   -2.19,   -3.12,   -6.29,   -8.78,  -10.36,  -13.39,
x   -13.37,    0.00/

 data  ((rmoqt_cnx(i,j), i = 1,37),j=1,10)/
x     0.00,    0.00,    0.00,    0.00,    0.00,    0.00,    0.00,
x     0.00,    0.00,    0.00,    0.00,    0.00,    0.00,    0.00,
x     0.00,    0.00,    0.00,    0.00,    0.00,    0.00,    0.00,
x     0.00,    0.00,    0.00,    0.00,    0.00,    0.00,    0.00,
x     0.00,    0.00,    0.00,    0.00,    0.00,    0.00,    0.00,
x     0.00,    0.00,
x     0.00,   -3.52,   -2.86,   -1.86,   -1.40,   -1.40,   -1.53,
x    -1.30,   -1.13,   -1.40,   -2.72,   -3.79,   -3.92,   -2.33,
x    -2.19,   -1.36,    0.33,   -0.10,   -0.13,   -2.79,   -3.75,
x    -4.95,   -3.85,   -1.89,   -0.37,    0.17,   -2.06,   -2.13,
x    -1.43,   -2.56,   -2.16,   -0.86,    0.73,    2.33,    4.28,
x     3.39,    0.00,
x     0.00,   -4.92,   -3.95,   -3.06,   -2.23,   -1.83,   -2.03,
x    -1.96,   -1.56,   -2.19,   -2.69,   -2.72,   -3.82,   -3.12,
x    -2.76,   -1.83,   -0.90,   -0.80,   -0.56,   -3.65,   -7.37,
x    -9.93,   -9.23,   -6.48,   -4.12,   -3.89,   -4.65,   -5.51,
x    -4.95,   -4.62,   -3.95,   -1.83,   -0.10,    2.69,    5.35,
x     4.92,    0.00,
x     0.00,   -5.12,   -4.38,   -3.89,   -2.29,   -2.19,   -2.23,
x    -2.16,   -1.93,   -4.02,   -5.38,   -4.92,   -4.78,   -4.85,
x    -4.68,   -3.32,   -2.36,   -2.13,   -2.66,   -9.60,  -15.08,
x   -16.64,  -14.15,   -9.86,   -7.71,   -8.70,   -8.54,   -7.31,
x    -7.11,   -4.65,   -3.09,   -1.66,    1.03,    3.59,    4.85,
x     4.68,    0.00,
x     0.00,   -4.75,   -4.52,   -4.19,   -3.55,   -3.26,   -4.72,
x    -4.98,   -4.55,   -5.05,   -6.05,   -5.51,   -5.95,   -5.68,
x    -5.75,   -5.02,   -4.52,   -3.99,   -6.11,  -14.98,  -21.06,
x   -21.92,  -18.53,  -14.61,  -14.25,  -11.96,  -10.40,   -9.50,
x    -8.24,   -6.21,   -4.48,   -2.36,    0.40,    3.02,    3.85,
x     2.59,    0.00,
x     0.00,   -4.48,   -5.68,   -4.82,   -5.08,   -6.48,   -6.78,
x    -6.08,   -6.58,   -7.61,   -7.34,   -5.78,   -7.04,   -5.81,
x    -5.81,   -6.74,   -7.27,   -7.71,  -12.72,  -20.46,  -26.57,
x   -25.87,  -23.62,  -20.33,  -18.73,  -15.68,  -13.25,  -11.09,
x    -8.80,   -7.37,   -5.68,   -3.32,    2.09,    6.24,    3.49,
x     3.85,    0.00,
x     0.00,   -4.78,   -8.04,   -6.68,   -6.14,   -8.07,   -8.70,
x    -6.74,   -6.78,   -7.37,   -7.44,   -6.88,   -7.04,   -7.01,
x    -7.24,   -7.81,   -9.57,  -11.43,  -18.67,  -24.21,  -30.06,
x   -27.47,  -24.81,  -24.91,  -23.15,  -18.83,  -15.98,  -13.75,
x   -11.36,   -9.33,   -5.71,   -2.69,   -0.50,    1.76,    3.09,
x     4.32,    0.00,
x     0.00,   -6.08,  -10.03,  -15.05,  -12.19,   -8.04,   -7.94,
x    -7.97,   -7.74,   -7.84,   -7.61,   -7.44,   -8.04,   -7.84,
x    -7.94,   -9.50,  -11.53,  -15.35,  -22.52,  -28.13,  -32.78,
x   -31.55,  -28.90,  -26.67,  -24.15,  -21.02,  -18.07,  -15.68,
x   -12.56,   -9.17,   -7.71,   -4.55,   -1.86,    1.73,    4.68,
x     7.34,    0.00,
x     0.00,   -3.89,   -7.41,  -11.79,  -10.50,   -7.84,   -6.81,
x    -7.31,   -7.67,   -8.97,   -8.74,   -9.00,   -9.80,   -8.90,
x    -9.37,  -10.79,  -17.17,  -17.60,  -25.67,  -29.53,  -33.81,
```

```
x   -34.24,  -29.03,  -24.41,  -23.22,  -20.93,  -18.10,  -15.91,
x   -13.45,  -11.13,   -9.67,   -7.17,   -3.75,   -0.07,    5.18,
x     3.59,    0.00,
x     0.00,   -3.62,   -5.68,   -7.54,   -9.13,   -5.91,   -5.25,
x    -5.65,   -6.88,   -9.63,  -10.36,  -10.60,  -10.96,  -10.40,
x   -10.70,  -11.82,  -18.37,  -18.73,  -25.91,  -27.63,  -28.17,
x   -27.60,  -25.11,  -20.29,  -19.16,  -18.90,  -17.77,  -15.94,
x   -14.12,  -12.65,  -11.66,   -9.33,   -5.98,   -0.10,    3.65,
x     2.16,    0.00/

 data  ((rmoqt_cnx(i,j), i = 1,37),j=11,20)/
x     0.00,   -2.69,   -2.69,   -2.29,   -3.59,   -5.48,   -5.15,
x    -6.64,   -5.98,   -8.04,   -9.13,   -9.43,  -10.60,  -11.36,
x   -10.66,  -13.05,  -17.04,  -16.04,  -22.88,  -23.42,  -20.63,
x   -18.00,  -16.94,  -14.91,  -15.74,  -15.91,  -16.57,  -14.75,
x   -13.75,  -13.22,  -11.99,   -9.47,   -6.14,   -0.47,    1.03,
x     0.13,    0.00,
x     0.00,   -0.90,   -0.13,    1.93,   -0.96,    0.80,   -3.62,
x    -4.52,   -7.17,   -6.64,   -8.00,   -9.50,  -10.23,   -9.57,
x    -9.23,  -14.12,  -10.56,  -11.49,  -17.80,  -15.78,  -13.45,
x   -11.56,   -9.53,   -8.84,   -9.93,  -11.56,  -12.36,  -12.52,
x   -13.09,  -11.92,  -10.93,   -9.50,   -7.54,   -4.28,   -3.52,
x    -3.69,    0.00,
x     0.00,   -0.83,   -0.17,    1.06,    1.73,    2.76,   -1.66,
x    -3.09,   -4.45,   -5.68,   -7.47,   -8.77,   -8.50,   -9.07,
x   -10.86,  -11.96,   -8.67,  -10.70,  -12.32,  -11.16,  -11.39,
x   -10.36,   -9.23,   -8.60,  -11.13,  -11.13,  -10.96,  -10.00,
x   -11.26,  -10.89,   -9.73,   -9.10,   -5.38,   -4.95,   -5.85,
x    -2.52,    0.00,
x     0.00,   -0.43,   -0.10,   -1.03,    0.80,   -0.76,   -3.85,
x    -4.02,   -4.09,   -5.15,   -5.51,   -6.44,   -7.44,   -8.87,
x    -9.57,   -6.88,   -5.65,   -9.50,   -8.70,   -8.34,   -8.54,
x   -11.06,   -9.20,   -9.53,   -9.47,   -8.54,   -7.91,   -8.97,
x    -8.37,   -9.07,   -8.57,   -6.48,   -5.75,   -2.66,   -3.69,
x    -1.20,    0.00,
x     0.00,   -0.53,   -0.93,   -0.66,   -1.79,   -2.46,   -2.46,
x    -2.69,   -3.09,   -3.29,   -3.35,   -4.02,   -6.21,   -7.57,
x    -4.98,   -1.79,   -3.59,   -4.98,   -5.08,   -6.61,   -6.91,
x    -6.34,   -8.34,   -8.44,   -6.91,   -5.65,   -6.14,   -6.84,
x    -6.84,   -4.78,   -5.12,   -5.38,   -2.89,   -2.72,   -2.13,
x    -0.86,    0.00,
x     0.00,   -0.33,   -0.70,   -0.73,   -0.20,   -1.03,   -2.66,
x    -1.56,   -0.47,   -0.90,   -2.26,   -4.15,   -2.69,   -1.20,
x     0.03,   -0.63,   -1.86,   -2.49,   -2.26,   -4.38,   -4.58,
x    -5.02,   -3.92,   -4.15,   -5.88,   -4.52,   -2.86,   -2.86,
x    -3.16,   -3.99,   -3.42,   -2.66,   -2.72,   -2.86,   -1.76,
x    -0.76,    0.00,
x     0.00,    0.56,    1.13,    1.30,    1.16,    1.36,    1.40,
x     1.13,    1.59,    0.50,    1.99,    0.40,    0.30,    0.33,
x    -0.33,    0.47,    1.20,    1.06,    1.30,    1.96,    2.26,
x     1.73,    1.13,   -0.23,    0.27,    0.27,    1.26,   -0.90,
x     0.33,   -1.36,   -1.69,   -2.06,   -1.23,   -1.06,   -1.10,
x    -0.37,    0.00,
x     0.00,    0.37,    0.47,    0.63,    1.33,    1.36,    1.86,
x     1.96,    2.16,    1.89,    1.86,    1.63,    1.49,    1.53,
x     1.26,    1.79,    1.56,    1.46,    2.13,    1.56,    2.03,
x     2.69,    2.86,    2.49,    2.79,    2.33,    1.66,    1.06,
```

```
     x     1.36,    1.26,    0.90,    1.36,    1.13,    0.80,    0.80,
     x     0.50,    0.00,
     x     0.00,   -0.07,   -0.17,   -0.07,    0.10,    0.00,    0.47,
     x     0.37,    0.40,    0.40,    0.10,   -0.33,   -0.66,   -0.47,
     x    -0.80,   -0.47,   -0.23,   -0.03,    0.00,    0.03,    0.23,
     x     0.47,    0.80,    0.47,    0.66,    0.33,   -0.10,   -0.40,
     x    -0.40,   -0.37,   -0.47,    0.00,   -0.10,    0.07,    0.17,
     x     0.07,    0.00,
     x     0.00,   -0.50,   -0.80,   -0.80,   -1.13,   -1.36,   -0.90,
     x    -1.26,   -1.36,   -1.06,   -1.66,   -2.33,   -2.79,   -2.49,
     x    -2.86,   -2.69,   -2.03,   -1.56,   -2.13,   -1.46,   -1.56,
     x    -1.79,   -1.26,   -1.53,   -1.49,   -1.63,   -1.86,   -1.89,
     x    -2.16,   -1.96,   -1.86,   -1.36,   -1.33,   -0.63,   -0.47,
     x    -0.37,    0.00/

      data  ((rmoqt_cnx(i,j), i = 1,37),j=21,30)/
     x     0.00,    0.37,    1.10,    1.06,    1.23,    2.06,    1.69,
     x     1.36,   -0.33,    0.90,   -1.26,   -0.27,   -0.27,    0.23,
     x    -1.13,   -1.73,   -2.26,   -1.96,   -1.30,   -1.06,   -1.20,
     x    -0.47,    0.33,   -0.33,   -0.30,   -0.40,   -1.99,   -0.50,
     x    -1.59,   -1.13,   -1.40,   -1.36,   -1.16,   -1.30,   -1.13,
     x    -0.56,    0.00,
     x     0.00,    0.76,    1.76,    2.86,    2.72,    2.66,    3.42,
     x     3.99,    3.16,    2.86,    2.86,    4.52,    5.88,    4.15,
     x     3.92,    5.02,    4.58,    4.38,    2.26,    2.49,    1.86,
     x     0.63,   -0.03,    1.20,    2.69,    4.15,    2.26,    0.90,
     x     0.47,    1.56,    2.66,    1.03,    0.20,    0.73,    0.70,
     x     0.33,    0.00,
     x     0.00,    0.86,    2.13,    2.72,    2.89,    5.38,    5.12,
     x     4.78,    6.84,    6.84,    6.14,    5.65,    6.91,    8.44,
     x     8.34,    6.34,    6.91,    6.61,    5.08,    4.98,    3.59,
     x     1.79,    4.98,    7.57,    6.21,    4.02,    3.35,    3.29,
     x     3.09,    2.69,    2.46,    2.46,    1.79,    0.66,    0.93,
     x     0.53,    0.00,
     x     0.00,    1.20,    3.69,    2.66,    5.75,    6.48,    8.57,
     x     9.07,    8.37,    8.97,    7.91,    8.54,    9.47,    9.53,
     x     9.20,   11.06,    8.54,    8.34,    8.70,    9.50,    5.65,
     x     6.88,    9.57,    8.87,    7.44,    6.44,    5.51,    5.15,
     x     4.09,    4.02,    3.85,    0.76,   -0.80,    1.03,    0.10,
     x     0.43,    0.00,
     x     0.00,    2.52,    5.85,    4.95,    5.38,    9.10,    9.73,
     x    10.89,   11.26,   10.00,   10.96,   11.13,   11.13,    8.60,
     x     9.23,   10.36,   11.39,   11.16,   12.32,   10.70,    8.67,
     x    11.96,   10.86,    9.07,    8.50,    8.77,    7.47,    5.68,
     x     4.45,    3.09,    1.66,   -2.76,   -1.73,   -1.06,    0.17,
     x     0.83,    0.00,
     x     0.00,    3.69,    3.52,    4.28,    7.54,    9.50,   10.93,
     x    11.92,   13.09,   12.52,   12.36,   11.56,    9.93,    8.84,
     x     9.53,   11.56,   13.45,   15.78,   17.80,   11.49,   10.56,
     x    14.12,    9.23,    9.57,   10.23,    9.50,    8.00,    6.64,
     x     7.17,    4.52,    3.62,   -0.80,    0.96,   -1.93,    0.13,
     x     0.90,    0.00,
     x     0.00,   -0.13,   -1.03,    0.47,    6.14,    9.47,   11.99,
     x    13.22,   13.75,   14.75,   16.57,   15.91,   15.74,   14.91,
     x    16.94,   18.00,   20.63,   23.42,   22.88,   16.04,   17.04,
     x    13.05,   10.66,   11.36,   10.60,    9.43,    9.13,    8.04,
     x     5.98,    6.64,    5.15,    5.48,    3.59,    2.29,    2.69,
```

```
x     2.69,    0.00,
x     0.00,   -2.16,   -3.65,    0.10,    5.98,    9.33,   11.66,
x    12.65,   14.12,   15.94,   17.77,   18.90,   19.16,   20.29,
x    25.11,   27.60,   28.17,   27.63,   25.91,   18.73,   18.37,
x    11.82,   10.70,   10.40,   10.96,   10.60,   10.36,    9.63,
x     6.88,    5.65,    5.25,    5.91,    9.13,    7.54,    5.68,
x     3.62,    0.00,
x     0.00,   -3.59,   -5.18,    0.07,    3.75,    7.17,    9.67,
x    11.13,   13.45,   15.91,   18.10,   20.93,   23.22,   24.41,
x    29.03,   34.24,   33.81,   29.53,   25.67,   17.60,   17.17,
x    10.79,    9.37,    8.90,    9.80,    9.00,    8.74,    8.97,
x     7.67,    7.31,    6.81,    7.84,   10.50,   11.79,    7.41,
x     3.89,    0.00,
x     0.00,   -7.34,   -4.68,   -1.73,    1.86,    4.55,    7.71,
x     9.17,   12.56,   15.68,   18.07,   21.02,   24.15,   26.67,
x    28.90,   31.55,   32.78,   28.13,   22.52,   15.35,   11.53,
x     9.50,    7.94,    7.84,    8.04,    7.44,    7.61,    7.84,
x     7.74,    7.97,    7.94,    8.04,   12.19,   15.05,   10.03,
x     6.08,    0.00/

 data  ((rmoqt_cnx(i,j), i = 1,37),j=31,37)/
x     0.00,   -4.32,   -3.09,   -1.76,    0.50,    2.69,    5.71,
x     9.33,   11.36,   13.75,   15.98,   18.83,   23.15,   24.91,
x    24.81,   27.47,   30.06,   24.21,   18.67,   11.43,    9.57,
x     7.81,    7.24,    7.01,    7.04,    6.88,    7.44,    7.37,
x     6.78,    6.74,    8.70,    8.07,    6.14,    6.68,    8.04,
x     4.78,    0.00,
x     0.00,   -3.85,   -3.49,   -6.24,   -2.09,    3.32,    5.68,
x     7.37,    8.80,   11.09,   13.25,   15.68,   18.73,   20.33,
x    23.62,   25.87,   26.57,   20.46,   12.72,    7.71,    7.27,
x     6.74,    5.81,    5.81,    7.04,    5.78,    7.34,    7.61,
x     6.58,    6.08,    6.78,    6.48,    5.08,    4.82,    5.68,
x     4.48,    0.00,
x     0.00,   -2.59,   -3.85,   -3.02,   -0.40,    2.36,    4.48,
x     6.21,    8.24,    9.50,   10.40,   11.96,   14.25,   14.61,
x    18.53,   21.92,   21.06,   14.98,    6.11,    3.99,    4.52,
x     5.02,    5.75,    5.68,    5.95,    5.51,    6.05,    5.05,
x     4.55,    4.98,    4.72,    3.26,    3.55,    4.19,    4.52,
x     4.75,    0.00,
x     0.00,   -4.68,   -4.85,   -3.59,   -1.03,    1.66,    3.09,
x     4.65,    7.11,    7.31,    8.54,    8.70,    7.71,    9.86,
x    14.15,   16.64,   15.08,    9.60,    2.66,    2.13,    2.36,
x     3.32,    4.68,    4.85,    4.78,    4.92,    5.38,    4.02,
x     1.93,    2.16,    2.23,    2.19,    2.29,    3.89,    4.38,
x     5.12,    0.00,
x     0.00,   -4.92,   -5.35,   -2.69,    0.10,    1.83,    3.95,
x     4.62,    4.95,    5.51,    4.65,    3.89,    4.12,    6.48,
x     9.23,    9.93,    7.37,    3.65,    0.56,    0.80,    0.90,
x     1.83,    2.76,    3.12,    3.82,    2.72,    2.69,    2.19,
x     1.56,    1.96,    2.03,    1.83,    2.23,    3.06,    3.95,
x     4.92,    0.00,
x     0.00,   -3.39,   -4.28,   -2.33,   -0.73,    0.86,    2.16,
x     2.56,    1.43,    2.13,    2.06,   -0.17,    0.37,    1.89,
x     3.85,    4.95,    3.75,    2.79,    0.13,    0.10,   -0.33,
x     1.36,    2.19,    2.33,    3.92,    3.79,    2.72,    1.40,
x     1.13,    1.30,    1.53,    1.40,    1.40,    1.86,    2.86,
x     3.52,    0.00,
```

```
x     0.00,   -1.33,   -1.63,   -2.09,   -1.66,   -2.03,   -2.06,
x    -1.83,   -0.43,   -0.03,   -0.73,   -0.90,   -1.20,   -0.66,
x    -0.30,   -0.23,    0.40,   -0.03,    0.03,   -0.17,   -0.27,
x     0.07,    0.23,    0.47,    1.26,    0.80,    0.53,    0.13,
x     0.03,   -0.76,   -0.66,   -0.63,   -0.37,   -0.33,    0.03,
x     0.40,    0.00/

 data  ((pmoqt_cnx(i,j), i = 1,37),j=1,10)/
x     0.00,  -26.87,  -32.72,  -37.60,  -36.10,  -32.58,  -27.14,
x   -21.56,  -18.67,  -13.75,   -8.27,   -6.44,   -0.50,    3.55,
x     7.57,   10.53,    6.64,    2.39,  -23.78,  -34.94,  -48.23,
x   -60.58,  -71.35,  -69.82,  -60.19,  -47.26,  -38.30,  -25.71,
x   -16.67,   -5.78,    2.26,   19.93,   34.14,   43.48,   33.88,
x    18.37,    0.00,
x     0.00,  -27.07,  -31.26,  -36.10,  -35.04,  -31.59,  -26.37,
x   -20.06,  -18.37,  -14.81,   -9.53,   -9.00,   -3.22,    0.76,
x     4.88,    7.64,    3.59,    0.03,  -17.97,  -35.91,  -50.12,
x   -62.44,  -71.05,  -68.65,  -59.45,  -47.50,  -36.60,  -28.33,
x   -18.83,   -9.17,    3.09,   16.97,   28.56,   35.97,   33.98,
x    19.03,    0.00,
x     0.00,  -24.98,  -29.63,  -33.48,  -32.58,  -29.26,  -24.08,
x   -17.77,  -16.54,  -15.71,  -10.20,   -8.40,   -3.99,    0.33,
x     3.29,    5.65,    1.36,   -1.73,  -12.75,  -37.60,  -53.38,
x   -62.74,  -68.02,  -65.20,  -55.67,  -43.74,  -36.40,  -26.77,
x   -18.83,  -10.56,    1.59,   14.55,   26.11,   33.41,   31.32,
x    21.36,    0.00,
x     0.00,  -21.59,  -25.94,  -29.33,  -28.43,  -26.34,  -21.12,
x   -16.31,  -16.28,  -14.61,   -7.91,   -7.01,   -3.52,   -0.23,
x     0.76,    2.92,   -1.00,   -4.55,  -14.98,  -40.16,  -55.50,
x   -63.07,  -65.80,  -61.48,  -50.82,  -40.75,  -33.88,  -26.34,
x   -16.77,  -10.50,   -0.96,   11.76,   22.88,   30.13,   27.93,
x    19.00,    0.00,
x     0.00,  -16.84,  -23.22,  -26.24,  -25.21,  -23.48,  -19.56,
x   -15.71,  -15.84,  -13.22,   -8.84,   -7.54,   -3.49,   -1.03,
x    -0.03,    0.60,   -3.65,   -6.01,  -15.35,  -38.76,  -54.97,
x   -62.44,  -62.94,  -56.90,  -46.50,  -37.53,  -32.62,  -24.05,
x   -15.38,   -8.07,   -1.13,    9.13,   19.13,   26.24,   24.74,
x    16.81,    0.00,
x     0.00,  -12.02,  -20.76,  -23.05,  -21.76,  -20.66,  -17.60,
x   -15.31,  -13.95,  -14.12,  -10.43,   -6.21,   -5.98,   -3.12,
x    -1.76,   -1.56,   -5.28,   -8.90,  -18.83,  -39.59,  -54.54,
x   -60.09,  -58.72,  -49.92,  -44.24,  -38.00,  -30.09,  -22.69,
x   -16.01,   -8.60,   -2.59,    1.69,   15.71,   25.34,   22.12,
x    16.94,    0.00,
x     0.00,   -5.58,  -16.61,  -20.26,  -18.97,  -18.83,  -15.81,
x   -14.15,  -12.62,  -11.43,   -8.30,   -6.14,   -5.31,   -4.62,
x    -3.95,   -3.19,   -7.57,  -10.53,  -20.06,  -38.93,  -51.52,
x   -57.89,  -55.30,  -47.40,  -41.72,  -34.44,  -28.53,  -21.59,
x   -14.75,   -7.84,   -2.46,    3.69,   11.79,   19.36,   21.02,
x    15.38,    0.00,
x     0.00,    3.35,   -6.31,   -8.77,  -11.99,  -15.91,  -14.15,
x   -11.99,  -11.09,  -10.96,   -8.80,   -7.11,   -5.85,   -4.62,
x    -5.75,   -5.68,  -10.70,  -13.19,  -21.39,  -37.17,  -48.26,
x   -51.28,  -48.06,  -42.75,  -37.30,  -31.35,  -26.04,  -20.36,
x   -15.11,  -10.30,   -4.42,    2.49,    7.67,   11.99,   13.25,
x     9.70,    0.00,
x     0.00,    5.25,    5.68,   -4.38,  -10.00,  -12.82,  -12.69,
```

```
x   -10.53,   -9.80,  -10.60,   -8.74,   -7.34,   -6.14,   -6.24,
x    -6.28,   -9.00,  -16.04,  -24.45,  -29.00,  -31.32,  -37.10,
x   -41.35,  -41.15,  -38.03,  -33.25,  -28.56,  -24.18,  -19.56,
x   -14.68,   -9.33,   -3.35,    2.96,    8.17,   11.46,    3.62,
x     1.96,    0.00,
x     0.00,    5.81,    9.37,   -1.40,   -5.21,  -11.82,  -12.16,
x   -10.79,   -9.47,  -10.46,   -7.87,   -6.24,   -6.31,   -6.88,
x    -7.37,  -10.50,  -19.23,  -33.65,  -33.78,  -30.92,  -28.40,
x   -33.85,  -35.27,  -33.98,  -30.36,  -25.84,  -22.05,  -17.14,
x   -12.39,   -7.34,   -2.06,    3.12,    7.54,    5.91,   -2.59,
x    -3.16,    0.00/

 data  ((pmoqt_cnx(i,j), i = 1,37),j=11,20)/
x     0.00,    5.05,    2.72,   -6.08,   -7.77,  -10.96,  -11.43,
x    -8.90,   -8.54,   -8.30,   -7.41,   -5.75,   -5.02,   -5.21,
x    -7.37,   -7.74,  -16.84,  -31.32,  -30.23,  -27.80,  -31.85,
x   -36.27,  -32.42,  -28.83,  -26.04,  -21.02,  -18.43,  -12.75,
x   -10.20,   -4.32,   -0.27,    3.79,    4.75,    1.83,   -1.96,
x    -2.89,    0.00,
x     0.00,    1.73,   -5.91,  -16.77,  -14.12,  -14.75,   -9.23,
x    -6.91,   -5.95,   -5.05,   -4.65,   -1.53,   -2.33,   -4.15,
x    -6.34,   -2.46,  -12.29,  -17.11,  -26.11,  -36.74,  -39.39,
x   -38.00,  -33.25,  -28.53,  -21.82,  -18.53,  -13.62,   -8.40,
x    -5.71,   -0.17,    2.13,    6.31,    7.77,    8.84,   12.32,
x     9.20,    0.00,
x     0.00,   -3.52,   -9.17,  -18.47,  -19.20,  -16.18,  -13.52,
x   -10.00,   -7.74,   -3.79,   -1.33,    0.23,    0.50,   -2.13,
x     1.89,    1.23,   -6.11,  -10.70,  -25.04,  -32.19,  -35.54,
x   -34.14,  -31.45,  -26.14,  -19.80,  -16.67,  -11.76,   -6.64,
x    -2.06,    1.03,    4.48,    5.05,    6.08,    9.70,   17.40,
x    10.93,    0.00,
x     0.00,   -3.72,  -11.89,  -14.81,  -18.47,  -16.31,  -11.59,
x   -10.30,   -7.94,   -3.72,   -0.07,    1.59,    2.33,    2.72,
x     2.62,   -0.83,   -4.09,   -4.22,  -18.33,  -24.48,  -28.70,
x   -28.03,  -27.00,  -21.99,  -15.64,  -12.99,   -9.43,   -5.38,
x     0.37,    3.39,    4.78,    6.05,    6.38,    7.64,   11.92,
x     8.00,    0.00,
x     0.00,   -2.29,   -7.97,   -9.83,  -12.85,  -12.59,   -8.67,
x    -4.95,   -5.78,   -4.02,   -1.96,    0.03,    3.49,    3.22,
x    -1.20,   -9.23,   -5.45,   -6.71,  -12.42,  -13.98,  -20.29,
x   -18.47,  -20.36,  -17.04,  -10.60,   -5.78,   -5.12,   -2.19,
x     0.33,    0.20,    1.93,    4.92,    5.38,    6.28,    6.31,
x     3.79,    0.00,
x     0.00,   -1.69,   -2.62,   -5.88,   -5.05,   -4.82,   -5.08,
x    -3.22,    0.27,    1.36,    2.36,    1.93,   -3.95,   -6.31,
x   -10.13,  -11.43,   -5.12,   -5.15,   -7.24,   -6.24,   -7.21,
x   -10.76,   -8.60,   -8.07,   -6.64,   -3.82,   -3.02,   -2.33,
x    -1.76,   -1.49,   -0.70,    0.86,    3.82,    4.82,    2.89,
x     1.69,    0.00,
x     0.00,   -0.93,   -1.00,   -2.49,   -2.16,   -2.99,   -2.03,
x    -1.83,   -2.46,   -1.26,   -6.21,   -5.08,   -5.95,   -6.44,
x    -1.43,   -2.86,   -5.78,   -6.98,   -9.00,  -10.23,  -10.73,
x   -11.03,  -10.93,   -8.54,   -8.07,   -7.27,   -6.34,   -3.99,
x    -3.35,   -0.83,    0.37,    1.89,    2.86,    4.02,    5.31,
x     1.36,    0.00,
x     0.00,   -1.83,   -2.29,   -2.99,   -5.31,   -2.36,   -6.58,
x    -5.12,   -4.09,   -3.89,   -3.16,   -1.13,    0.33,   -1.33,
```

```
x     1.16,   -0.43,    0.27,    0.56,   -5.38,    0.60,   -3.79,
x    -6.54,   -7.64,   -6.88,   -9.37,   -8.00,   -7.67,   -5.65,
x    -4.88,   -3.59,   -3.89,   -1.79,   -2.69,   -1.86,   -1.76,
x     0.40,    0.00,
x     0.00,   -0.73,   -2.03,   -2.42,   -4.02,   -2.06,   -5.25,
x    -4.35,   -4.48,   -4.78,   -5.41,   -4.58,   -4.52,   -4.12,
x    -3.26,   -3.49,   -1.76,    0.60,   -0.37,    0.60,   -1.76,
x    -3.49,   -3.26,   -4.12,   -4.52,   -4.58,   -5.41,   -4.78,
x    -4.48,   -4.35,   -5.25,   -2.06,   -4.02,   -2.42,   -2.03,
x    -0.73,    0.00,
x     0.00,    0.40,   -1.76,   -1.86,   -2.69,   -1.79,   -3.89,
x    -3.59,   -4.88,   -5.65,   -7.67,   -8.00,   -9.37,   -6.88,
x    -7.64,   -6.54,   -3.79,    0.60,   -5.38,    0.56,    0.27,
x    -0.43,    1.16,   -1.33,    0.33,   -1.13,   -3.16,   -3.89,
x    -4.09,   -5.12,   -6.58,   -2.36,   -5.31,   -2.99,   -2.29,
x    -1.83,    0.00/

 data  ((pmoqt_cnx(i,j), i = 1,37),j=21,30)/
x     0.00,    1.36,    5.31,    4.02,    2.86,    1.89,    0.37,
x    -0.83,   -3.35,   -3.99,   -6.34,   -7.27,   -8.07,   -8.54,
x   -10.93,  -11.03,  -10.73,  -10.23,   -9.00,   -6.98,   -5.78,
x    -2.86,   -1.43,   -6.44,   -5.95,   -5.08,   -6.21,   -1.26,
x    -2.46,   -1.83,   -2.03,   -2.99,   -2.16,   -2.49,   -1.00,
x    -0.93,    0.00,
x     0.00,    1.69,    2.89,    4.82,    3.82,    0.86,   -0.70,
x    -1.49,   -1.76,   -2.33,   -3.02,   -3.82,   -6.64,   -8.07,
x    -8.60,  -10.76,   -7.21,   -6.24,   -7.24,   -5.15,   -5.12,
x   -11.43,  -10.13,   -6.31,   -3.95,    1.93,    2.36,    1.36,
x     0.27,   -3.22,   -5.08,   -4.82,   -5.05,   -5.88,   -2.62,
x    -1.69,    0.00,
x     0.00,    3.79,    6.31,    6.28,    5.38,    4.92,    1.93,
x     0.20,    0.33,   -2.19,   -5.12,   -5.78,  -10.60,  -17.04,
x   -20.36,  -18.47,  -20.29,  -13.98,  -12.42,   -6.71,   -5.45,
x    -9.23,   -1.20,    3.22,    3.49,    0.03,   -1.96,   -4.02,
x    -5.78,   -4.95,   -8.67,  -12.59,  -12.85,   -9.83,   -7.97,
x    -2.29,    0.00,
x     0.00,    8.00,   11.92,    7.64,    6.38,    6.05,    4.78,
x     3.39,    0.37,   -5.38,   -9.43,  -12.99,  -15.64,  -21.99,
x   -27.00,  -28.03,  -28.70,  -24.48,  -18.33,   -4.22,   -4.09,
x    -0.83,    2.62,    2.72,    2.33,    1.59,   -0.07,   -3.72,
x    -7.94,  -10.30,  -11.59,  -16.31,  -18.47,  -14.81,  -11.89,
x    -3.72,    0.00,
x     0.00,   10.93,   17.40,    9.70,    6.08,    5.05,    4.48,
x     1.03,   -2.06,   -6.64,  -11.76,  -16.67,  -19.80,  -26.14,
x   -31.45,  -34.14,  -35.54,  -32.19,  -25.04,  -10.70,   -6.11,
x     1.23,    1.89,   -2.13,    0.50,    0.23,   -1.33,   -3.79,
x    -7.74,  -10.00,  -13.52,  -16.18,  -19.20,  -18.47,   -9.17,
x    -3.52,    0.00,
x     0.00,    9.20,   12.32,    8.84,    7.77,    6.31,    2.13,
x    -0.17,   -5.71,   -8.40,  -13.62,  -18.53,  -21.82,  -28.53,
x   -33.25,  -38.00,  -39.39,  -36.74,  -26.11,  -17.11,  -12.29,
x    -2.46,   -6.34,   -4.15,   -2.33,   -1.53,   -4.65,   -5.05,
x    -5.95,   -6.91,   -9.23,  -14.75,  -14.12,  -16.77,   -5.91,
x     1.73,    0.00,
x     0.00,   -2.89,   -1.96,    1.83,    4.75,    3.79,   -0.27,
x    -4.32,  -10.20,  -12.75,  -18.43,  -21.02,  -26.04,  -28.83,
x   -32.42,  -36.27,  -31.85,  -27.80,  -30.23,  -31.32,  -16.84,
```

```
x     -7.74,    -7.37,    -5.21,    -5.02,    -5.75,    -7.41,    -8.30,
x     -8.54,    -8.90,   -11.43,   -10.96,    -7.77,    -6.08,     2.72,
x      5.05,     0.00,
x      0.00,    -3.16,    -2.59,     5.91,     7.54,     3.12,    -2.06,
x     -7.34,   -12.39,   -17.14,   -22.05,   -25.84,   -30.36,   -33.98,
x    -35.27,   -33.85,   -28.40,   -30.92,   -33.78,   -33.65,   -19.23,
x    -10.50,    -7.37,    -6.88,    -6.31,    -6.24,    -7.87,   -10.46,
x     -9.47,   -10.79,   -12.16,   -11.82,    -5.21,    -1.40,     9.37,
x      5.81,     0.00,
x      0.00,     1.96,     3.62,    11.46,     8.17,     2.96,    -3.35,
x     -9.33,   -14.68,   -19.56,   -24.18,   -28.56,   -33.25,   -38.03,
x    -41.15,   -41.35,   -37.10,   -31.32,   -29.00,   -24.45,   -16.04,
x     -9.00,    -6.28,    -6.24,    -6.14,    -7.34,    -8.74,   -10.60,
x     -9.80,   -10.53,   -12.69,   -12.82,   -10.00,    -4.38,     5.68,
x      5.25,     0.00,
x      0.00,     9.70,    13.25,    11.99,     7.67,     2.49,    -4.42,
x    -10.30,   -15.11,   -20.36,   -26.04,   -31.35,   -37.30,   -42.75,
x    -48.06,   -51.28,   -48.26,   -37.17,   -21.39,   -13.19,   -10.70,
x     -5.68,    -5.75,    -4.62,    -5.85,    -7.11,    -8.80,   -10.96,
x    -11.09,   -11.99,   -14.15,   -15.91,   -11.99,    -8.77,    -6.31,
x      3.35,     0.00/

 data  ((pmoqt_cnx(i,j), i = 1,37),j=31,37)/
x      0.00,    15.38,    21.02,    19.36,    11.79,     3.69,    -2.46,
x     -7.84,   -14.75,   -21.59,   -28.53,   -34.44,   -41.72,   -47.40,
x    -55.30,   -57.89,   -51.52,   -38.93,   -20.06,   -10.53,    -7.57,
x     -3.19,    -3.95,    -4.62,    -5.31,    -6.14,    -8.30,   -11.43,
x    -12.62,   -14.15,   -15.81,   -18.83,   -18.97,   -20.26,   -16.61,
x     -5.58,     0.00,
x      0.00,    16.94,    22.12,    25.34,    15.71,     1.69,    -2.59,
x     -8.60,   -16.01,   -22.69,   -30.09,   -38.00,   -44.24,   -49.92,
x    -58.72,   -60.09,   -54.54,   -39.59,   -18.83,    -8.90,    -5.28,
x     -1.56,    -1.76,    -3.12,    -5.98,    -6.21,   -10.43,   -14.12,
x    -13.95,   -15.31,   -17.60,   -20.66,   -21.76,   -23.05,   -20.76,
x    -12.02,     0.00,
x      0.00,    16.81,    24.74,    26.24,    19.13,     9.13,    -1.13,
x     -8.07,   -15.38,   -24.05,   -32.62,   -37.53,   -46.50,   -56.90,
x    -62.94,   -62.44,   -54.97,   -38.76,   -15.35,    -6.01,    -3.65,
x      0.60,    -0.03,    -1.03,    -3.49,    -7.54,    -8.84,   -13.22,
x    -15.84,   -15.71,   -19.56,   -23.48,   -25.21,   -26.24,   -23.22,
x    -16.84,     0.00,
x      0.00,    19.00,    27.93,    30.13,    22.88,    11.76,    -0.96,
x    -10.50,   -16.77,   -26.34,   -33.88,   -40.75,   -50.82,   -61.48,
x    -65.80,   -63.07,   -55.50,   -40.16,   -14.98,    -4.55,    -1.00,
x      2.92,     0.76,    -0.23,    -3.52,    -7.01,    -7.91,   -14.61,
x    -16.28,   -16.31,   -21.12,   -26.34,   -28.43,   -29.33,   -25.94,
x    -21.59,     0.00,
x      0.00,    21.36,    31.32,    33.41,    26.11,    14.55,     1.59,
x    -10.56,   -18.83,   -26.77,   -36.40,   -43.74,   -55.67,   -65.20,
x    -68.02,   -62.74,   -53.38,   -37.60,   -12.75,    -1.73,     1.36,
x      5.65,     3.29,     0.33,    -3.99,    -8.40,   -10.20,   -15.71,
x    -16.54,   -17.77,   -24.08,   -29.26,   -32.58,   -33.48,   -29.63,
x    -24.98,     0.00,
x      0.00,    19.03,    33.98,    35.97,    28.56,    16.97,     3.09,
x     -9.17,   -18.83,   -28.33,   -36.60,   -47.50,   -59.45,   -68.65,
x    -71.05,   -62.44,   -50.12,   -35.91,   -17.97,     0.03,     3.59,
x      7.64,     4.88,     0.76,    -3.22,    -9.00,    -9.53,   -14.81,
```

```
x   -18.37,  -20.06,  -26.37,  -31.59,  -35.04,  -36.10,  -31.26,
x   -27.07,    0.00,
x     0.00,   18.37,   33.88,   43.48,   34.14,   19.93,    2.26,
x    -5.78,  -16.67,  -25.71,  -38.30,  -47.26,  -60.19,  -69.82,
x   -71.35,  -60.58,  -48.23,  -34.94,  -23.78,    2.39,    6.64,
x    10.53,    7.57,    3.55,   -0.50,   -6.44,   -8.27,  -13.75,
x   -18.67,  -21.56,  -27.14,  -32.58,  -36.10,  -37.60,  -32.72,
x   -26.87,    0.00/

 data  ((ymoqt_cnx(i,j), i = 1,37),j=1,10)/
x     0.00,    0.00,    0.00,    0.00,    0.00,    0.00,    0.00,
x     0.00,    0.00,    0.00,    0.00,    0.00,    0.00,    0.00,
x     0.00,    0.00,    0.00,    0.00,    0.00,    0.00,    0.00,
x     0.00,    0.00,    0.00,    0.00,    0.00,    0.00,    0.00,
x     0.00,    0.00,    0.00,    0.00,    0.00,    0.00,    0.00,
x     0.00,    0.00,
x    27.40,   16.21,   14.81,   15.51,   13.42,   12.99,   13.85,
x    14.42,   13.02,   15.01,   15.18,   20.06,   18.27,   15.28,
x    15.64,   17.04,   12.72,   14.08,   13.68,   12.39,   14.32,
x    15.25,   17.04,   16.44,   12.99,   12.75,    7.57,    7.91,
x     7.97,   12.65,   12.56,   11.82,   11.79,   12.65,   11.03,
x     9.77,   11.82,
x    34.81,   27.40,   23.91,   24.28,   21.95,   20.93,   19.00,
x    19.80,   20.79,   20.39,   20.99,   22.49,   24.05,   22.88,
x    23.81,   25.31,   23.58,   26.60,   28.27,   26.90,   26.94,
x    25.24,   25.48,   22.09,   16.01,   13.62,    9.77,   10.63,
x    14.18,   17.50,   19.46,   21.22,   23.81,   23.48,   21.76,
x    23.52,   25.77,
x    39.23,   34.84,   31.45,   31.12,   29.69,   26.24,   23.05,
x    20.69,   20.76,   19.83,   19.96,   22.49,   23.65,   26.01,
x    27.73,   29.63,   28.53,   29.79,   32.95,   34.41,   33.48,
x    32.15,   29.99,   24.48,   19.66,   16.01,   12.46,   13.85,
x    14.75,   18.70,   20.99,   24.21,   27.24,   27.00,   28.00,
x    31.55,   34.21,
x    36.93,   35.04,   31.65,   31.06,   28.86,   25.91,   20.63,
x    19.73,   18.80,   17.70,   18.17,   19.40,   22.25,   24.68,
x    27.20,   29.96,   29.13,   31.52,   33.48,   36.24,   34.68,
x    33.51,   31.29,   26.90,   19.20,   15.78,   14.58,   14.18,
x    16.18,   19.16,   22.49,   25.14,   29.43,   31.02,   32.35,
x    35.84,   38.20,
x    29.63,   28.80,   26.01,   23.75,   22.09,   19.56,   18.37,
x    18.40,   17.87,   17.11,   17.87,   19.36,   19.70,   20.53,
x    22.39,   24.98,   24.48,   26.77,   29.66,   31.92,   28.70,
x    26.14,   23.72,   20.19,   18.00,   17.37,   14.55,   14.58,
x    15.18,   16.37,   17.17,   16.64,   23.22,   23.62,   27.14,
x    31.02,   32.22,
x    23.58,   21.89,   20.46,   18.67,   17.21,   16.34,   18.67,
x    16.11,   15.01,   14.75,   16.51,   17.64,   18.63,   17.80,
x    18.23,   19.13,   18.43,   21.95,   24.35,   26.24,   23.52,
x    23.62,   21.92,   17.37,   16.97,   15.21,   14.05,   12.59,
x    13.02,   15.08,   18.00,   15.98,   16.97,   18.43,   20.89,
x    24.81,   25.94,
x    17.44,   17.54,   17.70,   19.26,   17.64,   13.52,   12.65,
x    13.92,   12.75,   12.29,   13.62,   14.91,   13.65,   14.42,
x    16.01,   17.64,   16.84,   20.46,   21.95,   20.93,   18.67,
x    20.23,   17.30,   14.28,   12.26,   13.52,   11.86,   11.53,
x    12.36,   13.78,   12.69,   13.25,   15.74,   16.14,   15.41,
```

```
x    19.26,   21.09,
x    16.08,   15.28,   16.41,   17.87,   15.68,   10.86,    8.93,
x     9.57,   10.76,   10.96,   11.16,   10.20,    9.73,   10.89,
x    12.69,   14.25,   15.01,   18.30,   18.20,   16.87,   16.87,
x    17.44,   14.81,    9.96,    9.07,    9.40,    9.03,    9.50,
x     9.67,    9.67,    8.47,    9.30,   12.75,   13.88,   16.87,
x    18.30,   19.73,
x    11.82,   11.29,   11.49,   11.66,   12.46,    6.84,    5.31,
x     6.31,    6.68,    5.98,    7.24,    7.14,    6.68,    6.54,
x     8.00,    9.60,   10.03,   13.55,   13.25,   12.29,   10.03,
x    10.13,    9.00,    5.02,    4.75,    6.14,    4.82,    6.11,
x     5.98,    5.08,    4.35,    4.98,    8.07,   10.70,   13.42,
x    13.82,   14.68/

 data  ((ymoqt_cnx(i,j), i = 1,37),j=11,20)/
x     6.24,    6.11,    6.05,    3.89,    4.25,    1.93,    0.40,
x     0.27,    1.00,   -1.13,    1.10,    1.03,    2.09,    1.86,
x     4.45,    4.22,    5.68,    7.64,    8.14,    6.78,    4.12,
x     1.36,    1.53,   -2.56,   -2.89,   -2.82,   -0.13,   -1.30,
x     0.70,   -0.47,   -0.40,    0.90,    4.42,    5.88,    7.17,
x     6.61,    7.01,
x    -2.13,   -1.30,   -3.85,   -5.68,   -4.65,   -9.96,   -9.03,
x    -9.93,   -5.12,  -11.49,   -4.62,   -8.40,   -6.24,   -5.25,
x     0.20,   -4.88,   -3.59,   -1.73,   -2.52,   -2.92,   -6.68,
x    -7.71,   -8.57,  -13.32,  -12.69,  -13.42,   -9.03,  -12.95,
x    -6.94,   -9.10,   -7.67,   -7.11,   -2.49,   -4.42,   -4.88,
x    -3.29,   -3.39,
x   -10.56,  -10.06,  -11.63,  -13.25,  -17.17,  -16.67,  -20.89,
x   -17.14,  -17.40,  -22.55,  -18.40,  -17.07,  -18.70,  -11.29,
x   -15.31,  -12.06,  -12.16,  -12.56,  -12.29,  -13.65,  -14.95,
x   -15.74,  -20.53,  -19.20,  -23.72,  -19.86,  -21.06,  -24.18,
x   -19.56,  -18.10,  -20.53,  -12.72,  -13.52,  -10.73,  -14.12,
x   -11.09,  -11.36,
x   -20.53,  -18.43,  -22.72,  -21.42,  -24.91,  -29.86,  -29.93,
x   -30.33,  -32.45,  -26.87,  -32.55,  -30.96,  -29.16,  -28.56,
x   -23.15,  -20.23,  -23.08,  -21.52,  -22.98,  -22.75,  -26.24,
x   -25.18,  -28.46,  -32.98,  -31.79,  -33.12,  -34.74,  -29.23,
x   -35.11,  -33.55,  -31.16,  -27.53,  -21.95,  -20.00,  -24.55,
x   -21.42,  -21.89,
x   -34.51,  -33.28,  -32.98,  -36.07,  -35.01,  -35.94,  -41.65,
x   -45.01,  -42.61,  -41.75,  -43.61,  -48.43,  -45.11,  -36.17,
x   -31.95,  -34.84,  -30.52,  -32.42,  -35.57,  -32.58,  -33.91,
x   -38.70,  -35.11,  -37.70,  -43.41,  -47.86,  -45.14,  -44.28,
x   -44.64,  -47.00,  -42.45,  -35.91,  -33.05,  -37.37,  -33.08,
x   -34.41,  -34.38,
x   -43.31,  -43.78,  -44.04,  -43.91,  -46.53,  -46.37,  -45.14,
x   -47.80,  -53.08,  -53.87,  -53.77,  -53.71,  -46.67,  -46.23,
x   -46.14,  -43.48,  -46.60,  -46.37,  -45.84,  -46.70,  -46.30,
x   -46.14,  -49.95,  -49.39,  -47.30,  -49.49,  -52.51,  -52.38,
x   -51.75,  -49.72,  -45.17,  -46.67,  -47.07,  -45.17,  -45.14,
x   -45.01,  -43.84,
x   -41.42,  -46.97,  -48.03,  -49.82,  -51.15,  -49.26,  -48.29,
x   -49.19,  -40.42,  -44.54,  -36.47,  -43.48,  -44.91,  -46.73,
x   -45.14,  -44.81,  -44.31,  -46.14,  -46.07,  -47.63,  -48.09,
x   -47.76,  -49.02,  -51.81,  -50.92,  -48.96,  -43.74,  -47.70,
x   -44.94,  -48.86,  -49.86,  -50.19,  -48.13,  -47.56,  -47.63,
x   -47.26,  -47.23,
```

```
x   -18.33,   -4.95,   -8.57,  -10.33,  -13.42,  -21.82,  -17.34,
x   -25.14,  -26.84,  -35.17,  -32.38,  -33.81,  -29.13,  -31.29,
x   -25.04,  -24.28,  -21.89,  -24.65,  -23.32,  -23.38,  -23.88,
x   -26.24,  -30.23,  -30.03,  -36.37,  -38.23,  -41.75,  -43.05,
x   -42.32,  -39.63,  -40.59,  -36.20,  -37.00,  -35.21,  -36.70,
x   -35.51,  -33.51,
x     0.07,   15.28,   14.05,   12.46,   11.79,    7.21,   11.63,
x     7.24,    7.74,    3.92,    4.68,    2.23,    3.62,   -0.63,
x     2.59,    0.96,    1.00,   -0.63,    0.00,    0.63,   -1.00,
x    -0.96,   -2.59,    0.63,   -3.62,   -2.23,   -4.68,   -3.92,
x    -7.74,   -7.24,  -11.63,   -7.21,  -11.79,  -12.46,  -14.05,
x   -15.28,   -0.07,
x    33.51,   35.51,   36.70,   35.21,   37.00,   36.20,   40.59,
x    39.63,   42.32,   43.05,   41.75,   38.23,   36.37,   30.03,
x    30.23,   26.24,   23.88,   23.38,   23.32,   24.65,   21.89,
x    24.28,   25.04,   31.29,   29.13,   33.81,   32.38,   35.17,
x    26.84,   25.14,   17.34,   21.82,   13.42,   10.33,    8.57,
x     4.95,   18.33/

 data  ((ymoqt_cnx(i,j), i = 1,37),j=21,30)/
x    47.23,   47.26,   47.63,   47.56,   48.13,   50.19,   49.86,
x    48.86,   44.94,   47.70,   43.74,   48.96,   50.92,   51.81,
x    49.02,   47.76,   48.09,   47.63,   46.07,   46.14,   44.31,
x    44.81,   45.14,   46.73,   44.91,   43.48,   36.47,   44.54,
x    40.42,   49.19,   48.29,   49.26,   51.15,   49.82,   48.03,
x    46.97,   41.42,
x    43.84,   45.01,   45.14,   45.17,   47.07,   46.67,   45.17,
x    49.72,   51.75,   52.38,   52.51,   49.49,   47.30,   49.39,
x    49.95,   46.14,   46.30,   46.70,   45.84,   46.37,   46.60,
x    43.48,   46.14,   46.23,   46.67,   53.71,   53.77,   53.87,
x    53.08,   47.80,   45.14,   46.37,   46.53,   43.91,   44.04,
x    43.78,   43.31,
x    34.38,   34.41,   33.08,   37.37,   33.05,   35.91,   42.45,
x    47.00,   44.64,   44.28,   45.14,   47.86,   43.41,   37.70,
x    35.11,   38.70,   33.91,   32.58,   35.57,   32.42,   30.52,
x    34.84,   31.95,   36.17,   45.11,   48.43,   43.61,   41.75,
x    42.61,   45.01,   41.65,   35.94,   35.01,   36.07,   32.98,
x    33.28,   34.51,
x    21.89,   21.42,   24.55,   20.00,   21.95,   27.53,   31.16,
x    33.55,   35.11,   29.23,   34.74,   33.12,   31.79,   32.98,
x    28.46,   25.18,   26.24,   22.75,   22.98,   21.52,   23.08,
x    20.23,   23.15,   28.56,   29.16,   30.96,   32.55,   26.87,
x    32.45,   30.33,   29.93,   29.86,   24.91,   21.42,   22.72,
x    18.43,   20.53,
x    11.36,   11.09,   14.12,   10.73,   13.52,   12.72,   20.53,
x    18.10,   19.56,   24.18,   21.06,   19.86,   23.72,   19.20,
x    20.53,   15.74,   14.95,   13.65,   12.29,   12.56,   12.16,
x    12.06,   15.31,   11.29,   18.70,   17.07,   18.40,   22.55,
x    17.40,   17.14,   20.89,   16.67,   17.17,   13.25,   11.63,
x    10.06,   10.56,
x     3.39,    3.29,    4.88,    4.42,    2.49,    7.11,    7.67,
x     9.10,    6.94,   12.95,    9.03,   13.42,   12.69,   13.32,
x     8.57,    7.71,    6.68,    2.92,    2.52,    1.73,    3.59,
x     4.88,   -0.20,    5.25,    6.24,    8.40,    4.62,   11.49,
x     5.12,    9.93,    9.03,    9.96,    4.65,    5.68,    3.85,
x     1.30,    2.13,
x    -7.01,   -6.61,   -7.17,   -5.88,   -4.42,   -0.90,    0.40,
```

```
x     0.47,   -0.70,    1.30,    0.13,    2.82,    2.89,    2.56,
x    -1.53,   -1.36,   -4.12,   -6.78,   -8.14,   -7.64,   -5.68,
x    -4.22,   -4.45,   -1.86,   -2.09,   -1.03,   -1.10,    1.13,
x    -1.00,   -0.27,   -0.40,   -1.93,   -4.25,   -3.89,   -6.05,
x    -6.11,   -6.24,
x   -14.68,  -13.82,  -13.42,  -10.70,   -8.07,   -4.98,   -4.35,
x    -5.08,   -5.98,   -6.11,   -4.82,   -6.14,   -4.75,   -5.02,
x    -9.00,  -10.13,  -10.03,  -12.29,  -13.25,  -13.55,  -10.03,
x    -9.60,   -8.00,   -6.54,   -6.68,   -7.14,   -7.24,   -5.98,
x    -6.68,   -6.31,   -5.31,   -6.84,  -12.46,  -11.66,  -11.49,
x   -11.29,  -11.82,
x   -19.73,  -18.30,  -16.87,  -13.88,  -12.75,   -9.30,   -8.47,
x    -9.67,   -9.67,   -9.50,   -9.03,   -9.40,   -9.07,   -9.96,
x   -14.81,  -17.44,  -16.87,  -16.87,  -18.20,  -18.30,  -15.01,
x   -14.25,  -12.69,  -10.89,   -9.73,  -10.20,  -11.16,  -10.96,
x   -10.76,   -9.57,   -8.93,  -10.86,  -15.68,  -17.87,  -16.41,
x   -15.28,  -16.08,
x   -21.09,  -19.26,  -15.41,  -16.14,  -15.74,  -13.25,  -12.69,
x   -13.78,  -12.36,  -11.53,  -11.86,  -13.52,  -12.26,  -14.28,
x   -17.30,  -20.23,  -18.67,  -20.93,  -21.95,  -20.46,  -16.84,
x   -17.64,  -16.01,  -14.42,  -13.65,  -14.91,  -13.62,  -12.29,
x   -12.75,  -13.92,  -12.65,  -13.52,  -17.64,  -19.26,  -17.70,
x   -17.54,  -17.44/

 data  ((ymoqt_cnx(i,j), i = 1,37),j=31,37)/
x   -25.94,  -24.81,  -20.89,  -18.43,  -16.97,  -15.98,  -18.00,
x   -15.08,  -13.02,  -12.59,  -14.05,  -15.21,  -16.97,  -17.37,
x   -21.92,  -23.62,  -23.52,  -26.24,  -24.35,  -21.95,  -18.43,
x   -19.13,  -18.23,  -17.80,  -18.63,  -17.64,  -16.51,  -14.75,
x   -15.01,  -16.11,  -18.67,  -16.34,  -17.21,  -18.67,  -20.46,
x   -21.89,  -23.58,
x   -32.22,  -31.02,  -27.14,  -23.62,  -23.22,  -16.64,  -17.17,
x   -16.37,  -15.18,  -14.58,  -14.55,  -17.37,  -18.00,  -20.19,
x   -23.72,  -26.14,  -28.70,  -31.92,  -29.66,  -26.77,  -24.48,
x   -24.98,  -22.39,  -20.53,  -19.70,  -19.36,  -17.87,  -17.11,
x   -17.87,  -18.40,  -18.37,  -19.56,  -22.09,  -23.75,  -26.01,
x   -28.80,  -29.63,
x   -38.20,  -35.84,  -32.35,  -31.02,  -29.43,  -25.14,  -22.49,
x   -19.16,  -16.18,  -14.18,  -14.58,  -15.78,  -19.20,  -26.90,
x   -31.29,  -33.51,  -34.68,  -36.24,  -33.48,  -31.52,  -29.13,
x   -29.96,  -27.20,  -24.68,  -22.25,  -19.40,  -18.17,  -17.70,
x   -18.80,  -19.73,  -20.63,  -25.91,  -28.86,  -31.06,  -31.65,
x   -35.04,  -36.93,
x   -34.21,  -31.55,  -28.00,  -27.00,  -27.24,  -24.21,  -20.99,
x   -18.70,  -14.75,  -13.85,  -12.46,  -16.01,  -19.66,  -24.48,
x   -29.99,  -32.15,  -33.48,  -34.41,  -32.95,  -29.79,  -28.53,
x   -29.63,  -27.73,  -26.01,  -23.65,  -22.49,  -19.96,  -19.83,
x   -20.76,  -20.69,  -23.05,  -26.24,  -29.69,  -31.12,  -31.45,
x   -34.84,  -39.23,
x   -25.77,  -23.52,  -21.76,  -23.48,  -23.81,  -21.22,  -19.46,
x   -17.50,  -14.18,  -10.63,   -9.77,  -13.62,  -16.01,  -22.09,
x   -25.48,  -25.24,  -26.94,  -26.90,  -28.27,  -26.60,  -23.58,
x   -25.31,  -23.81,  -22.88,  -24.05,  -22.49,  -20.99,  -20.39,
x   -20.79,  -19.80,  -19.00,  -20.93,  -21.95,  -24.28,  -23.91,
x   -27.40,  -34.81,
x   -11.82,   -9.77,  -11.03,  -12.65,  -11.79,  -11.82,  -12.56,
x   -12.65,   -7.97,   -7.91,   -7.57,  -12.75,  -12.99,  -16.44,
```

```
     x   -17.04,  -15.25,  -14.32,  -12.39,  -13.68,  -14.08,  -12.72,
     x   -17.04,  -15.64,  -15.28,  -18.27,  -20.06,  -15.18,  -15.01,
     x   -13.02,  -14.42,  -13.85,  -12.99,  -13.42,  -15.51,  -14.81,
     x   -16.21,  -27.40,
     x     0.76,    2.82,    2.06,    3.65,   -0.17,    2.72,    0.47,
     x     1.20,    1.13,    0.80,    0.47,   -1.46,   -3.92,   -2.19,
     x    -1.20,   -0.93,   -0.73,   -0.40,   -0.33,   -0.37,   -0.17,
     x    -2.42,   -1.79,   -1.79,   -3.59,   -4.42,   -1.93,   -1.26,
     x    -2.26,   -2.69,   -2.16,   -1.63,   -2.09,   -2.69,   -1.56,
     x    -1.10,   -0.03/

C (alfao,betao) is the pt in the region [-90,90]x[0,180] with the same
aerodynamics as
C (alf2d,bet2d) except for signs.  The matrix "s" supplies the signs
for beta < 0.

      data s/
     x     1., 1., 1., 1., 1., 1.,
     x     1.,-1., 1.,-1., 1.,-1./

      va22  = va2s2(1)**2 + va2s2(2)**2 + va2s2(3)**2
      va2   = sqrt(va22)
      q2    = .5*rho*va22
      alf2  = 0.
      bet2  = 0.
      if (abs(va2s2(1))+abs(va2s2(3)).gt..0001)
     x      alf2 = atan(va2s2(3)/va2s2(1))
      calf  = cos(alf2)
      sbet  = 0
      cbet  = 1
      if (va2.gt.2) then
         sbet  = va2s2(2)/va2
         cbet  = va2s2(1)/(va2*calf)
         bet2 = atan2(sbet,cbet)
      end if
      alf2d = alf2*r2d
      bet2d = bet2*r2d

C  (alfao,betao) = equivalent (alf, beta) in [-90,90] x [0,90]
      alfao = alf2d
      betao = abs(bet2d)
      quad  = 1
      if (bet2d.lt.0) quad = 2

C aero components at (alfao, betao), betao > 0

      call serch1(alfao, alfg_cnx, 37, ix, sigx)
      call serch1(betao, betg_cnx, 37, iy, sigy)
      fawo(1)  = -q2*f2d(ix, iy, sigx, sigy, 37, doqt_cnx)
      fawo(2)  =  q2*f2d(ix, iy, sigx, sigy, 37, yoqt_cnx)
      fawo(3)  = -q2*f2d(ix, iy, sigx, sigy, 37, loqt_cnx)
      fawo(4)  =  q2*f2d(ix, iy, sigx, sigy, 37, rmoqt_cnx)
      fawo(5)  =  q2*f2d(ix, iy, sigx, sigy, 37, pmoqt_cnx)
      fawo(6)  =  q2*f2d(ix, iy, sigx, sigy, 37, ymoqt_cnx)
```

```
C transform from wind axes to body axes: Tbw  = E2(alf2)*E3(-bet2),
fa22 = Tbw*fa2w,  ma22 = Tbw*ma2w

      do 100 j = 1,6
 100  faw(j) = s(j,quad)*fawo(j)

      salf     =  sin(alf2)
      T2w(1,1) =  calf*cbet
      T2w(2,1) =  sbet
      T2w(3,1) =  salf*cbet
      T2w(1,2) = -calf*sbet
      T2w(2,2) =  cbet
      T2w(3,2) = -salf*sbet
      T2w(1,3) = -salf
      T2w(2,3) =  0.
      T2w(3,3) =  calf
      fa22(1) = T2w(1,1)*fa2w(1)+T2w(1,2)*fa2w(2)+T2w(1,3)*fa2w(3)
      fa22(2) = T2w(2,1)*fa2w(1)+T2w(2,2)*fa2w(2)+T2w(2,3)*fa2w(3)
      fa22(3) = T2w(3,1)*fa2w(1)+T2w(3,2)*fa2w(2)+T2w(3,3)*fa2w(3)
      ma22(1) = T2w(1,1)*ma2w(1)+T2w(1,2)*ma2w(2)+T2w(1,3)*ma2w(3)
      ma22(2) = T2w(2,1)*ma2w(1)+T2w(2,2)*ma2w(2)+T2w(2,3)*ma2w(3)
      ma22(3) = T2w(3,1)*ma2w(1)+T2w(3,2)*ma2w(2)+T2w(3,3)*ma2w(3)

      return
      end



c locate independent variable x in array of values, tx, for table
lookup

      SUBROUTINE SERCH1(x, tx, nx, ix, sigx)
      DIMENSION tx(1)
      x1 = AMAX1(tx(1),AMIN1(x,tx(nx)))
      i  = 1
 1    IF (i .EQ. nx) THEN
         ix   = nx - 1
         sigx = 1
      ELSE IF (x1 .LT. tx(i+1)) THEN
         ix   = i
         sigx = (x1 - tx(i))/(tx(i+1)-tx(i))
      ELSE
         i = i + 1
         GO TO 1
      END IF
      RETURN
      END

c  2-D table lookup routine.

      FUNCTION F2D(ix, iy, sigx, sigy, nx, tf)
      DIMENSION tf(1)
      k01 = nx*iy + ix
      k11 = k01 + 1
      k00 = k01 - nx
      k10 = k00 + 1
      f00 = tf(k00)
```

```
f01 = tf(k01)
f10 = tf(k10)
f11 = tf(k11)
sigys = 1 - sigx
IF (sigy .LE. sigys) THEN
   F2D = f00 + (f10 - f00)*sigx + (f01 - f00)*sigy
ELSE
   F2D = f11 - (f11 - f01)*(1 - sigx) - (f11 - f10)*(1 - sigy)
END IF
RETURN
END
```

# APPENDIX C SLINGTORQ.F

Sling Torque data from the Ames mechanical engineering laboratory test. Lookup program to extract required data follows the data. Program by Ehlers G.

```
C   /GenHel/sl/sling_dat.f  Created 01 Sep 2001 by George E Ehlers
C   Data for sling wind-up yaw resistance obtained from laboratory
tests

      subroutine sling_dat

      include 'slvars.cmn'

      real beta_cnx(110), torq_cnx(110)




C Beta values

      data beta_cnx/
     x          0,   0.1020,   0.2039,    0.3059,    0.4078,    0.5098,
     x     0.6118,   0.7137,   0.8157,    0.9176,    1.0196,    1.1216,
     x     1.2235,   1.3255,   1.4275,    1.5294,    1.6314,    1.7333,
     x     1.8353,   1.9373,   2.0392,    2.1412,    2.2431,    2.3451,
     x     2.4471,   2.5490,   2.6510,    2.7529,    2.8549,    2.9569,
     x     3.0588,   3.1608,   3.2627,    3.3647,    3.4667,
3.5686,
     x     3.6706,   3.7725,   3.8745,    3.9765,    4.0784,    4.1804,
     x     4.2824,   4.3843,   4.4863,    4.5882,    4.6902,    4.7922,
     x     4.8941,   4.9961,   5.0980,    5.2000,    5.2100,    6.2204,
     x     7.2307,   8.2411,   9.2514,   10.2618,   11.2721,   12.2825,
     x    13.2928,  14.3032,  15.3135,   16.3239,   17.3342,   18.3446,
     x    19.3549,  20.3653,  21.3756,   22.3860,   23.3963,   24.4067,
     x    25.4170,  26.4274,  27.4377,   28.4481,   29.4584,   30.4688,
     x    31.4791,  32.4895,  33.4998,   34.5102,   35.5205,   36.5309,
     x    37.5412,  38.5516,  39.5619,   40.5723,   41.5826,   42.5930,
     x    43.6033,  44.6137,  45.6240,   46.6344,   47.6447,   48.6551,
     x    49.6654,  50.6758,  51.6861,   52.6965,   53.7068,   54.7172,
     x    55.7275,  56.7379,  57.7482,   58.7586,   59.7689,   60.7793,
     x    61.7896,  62.8000/

C123456789C123456789C123456789C123456789C123456789C123456789C1234567891
2
C Force values

      data torq_cnx/
     x    0.5187,    0.9107,    0.6239,    1.0651,    2.0096,
2.5485,
     x    2.0417,    2.0745,    3.5813,    9.0015,   16.1410,
24.8609,
```

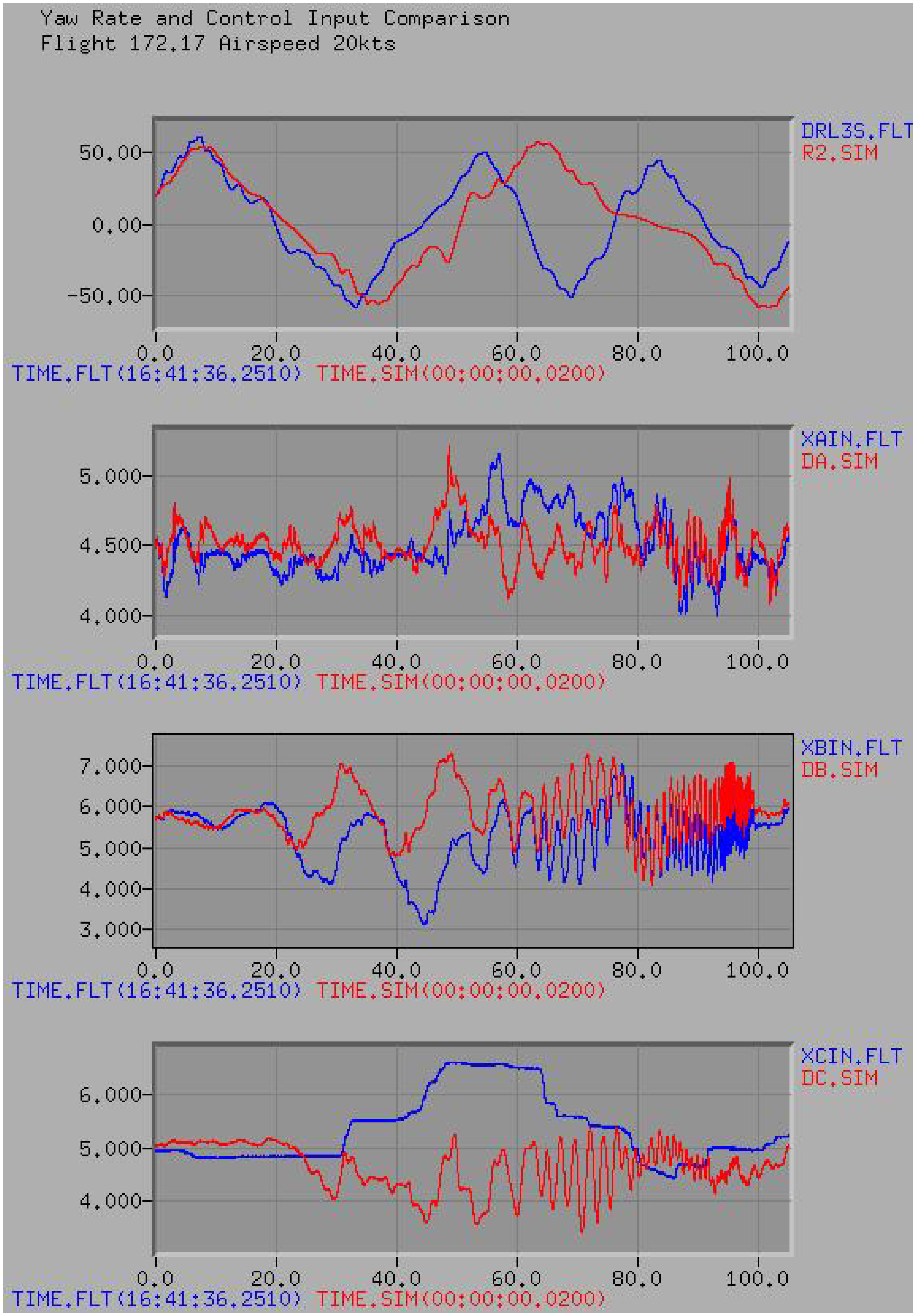
```
     x   34.9526,   46.1588,   58.1917,   70.7486,   83.5244,
96.2232,
     x  108.5666,  120.3007,  131.2011,  141.0772,  149.7737,
157.1724,
     x  163.1922,  167.7876,  170.9482,  172.6952,  173.0795,
172.1783,
     x  170.0910,  166.9360,  162.8468,  157.9675,  152.4495,
146.4473,
     x   40.1151,  133.6033,  127.0553,  120.6046,  114.3726,
108.4663,
     x  102.9770,   97.9784,   93.5264,   89.6585,   86.3940,
83.7344,
     x   81.6643,   80.1529,   79.1556,   78.6159,   78.3361,
79.4923,
     x   80.6913,   81.9331,   83.2176,   84.5449,   85.9150,
87.3278,
     x   88.7834,   90.2818,   91.8229,   93.4068,   95.0335,
96.7029,
     x   98.4151,  100.1701,  101.9679,  103.8084,  105.6917,
107.6177,
     x  109.5865,  111.5981,  113.6525,  115.7496,  117.8895,
120.0721,
     x  122.2976,  124.5658,  126.8767,  129.2305,  131.6270,
134.066,
     x  136.5483,  139.0731,  141.6407,  144.2510,  146.9041,
149.6000,
     x  152.3386,  155.1201,  157.9442,  160.8112,  163.7209,
166.6734,
     x  169.6687,  172.7067,  175.7875,  178.9111,  182.0774,
185.2865,
     x  188.5384,  191.8330,  195.1704,  198.5506,  201.9735,
205.4392,
     x  208.9477,  212.4989/

c
      beta=abs(ps2-ps1)

      call serch1(beta, beta_cnx, 110, ix, sigx)

      kk1=-sign(1,(ps2-ps1))
      kk2=sigx*((torq_cnx(ix+1)-
     x       torq_cnx(ix))/(beta_cnx(ix+1)-beta_cnx(ix)))
      Kps2= -sign(1,(ps2-ps1))*(torq_cnx(ix)+sigx*((torq_cnx(ix+1)-
     x       torq_cnx(ix))

      RETURN
      END


      SUBROUTINE SERCH1(x, tx, nx, ix, sigx)
      DIMENSION tx(1)
      x1 = AMAX1(tx(1),AMIN1(x,tx(nx)))
      i  = 1
 1    IF (i .EQ. nx) THEN
         ix   = nx - 1
         sigx = 1
      ELSE IF (x1 .LT. tx(i+1)) THEN
```

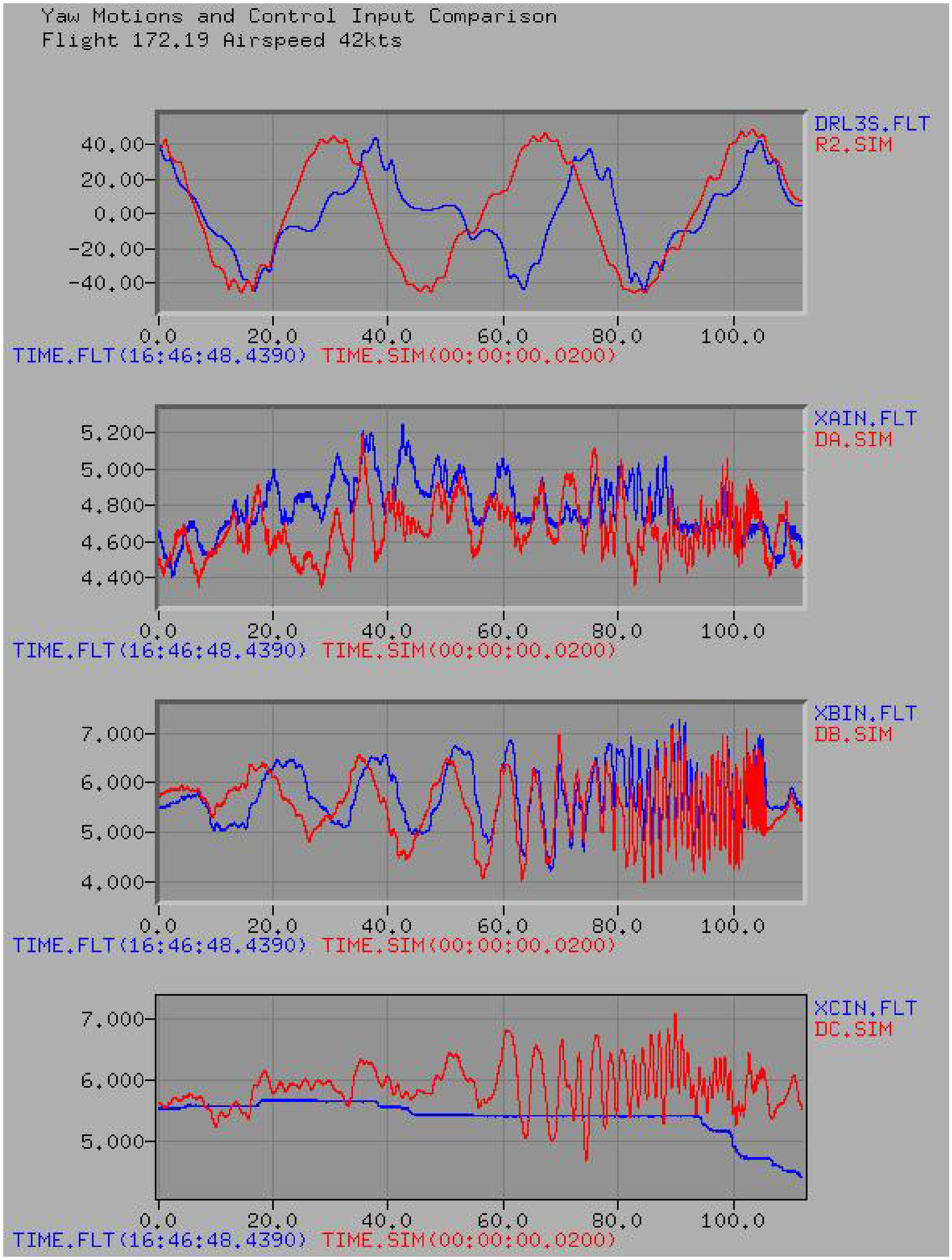
```
   ix   = i
   sigx = (x1 - tx(i))/(tx(i+1)-tx(i))
ELSE
   i = i + 1
   GO TO 1
END IF
RETURN
END
```

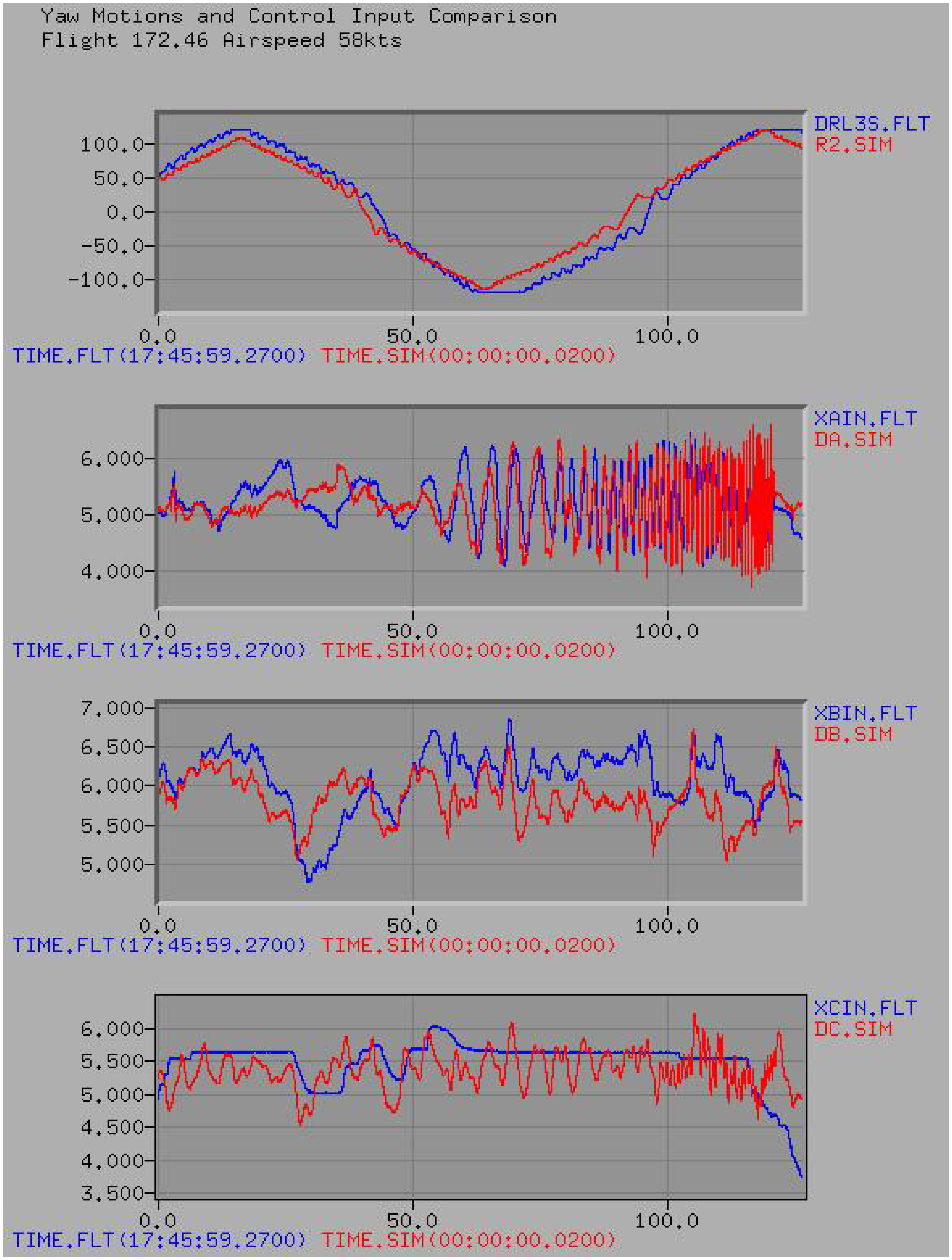
THIS PAGE INTENTIONALLY LEFT BLANK

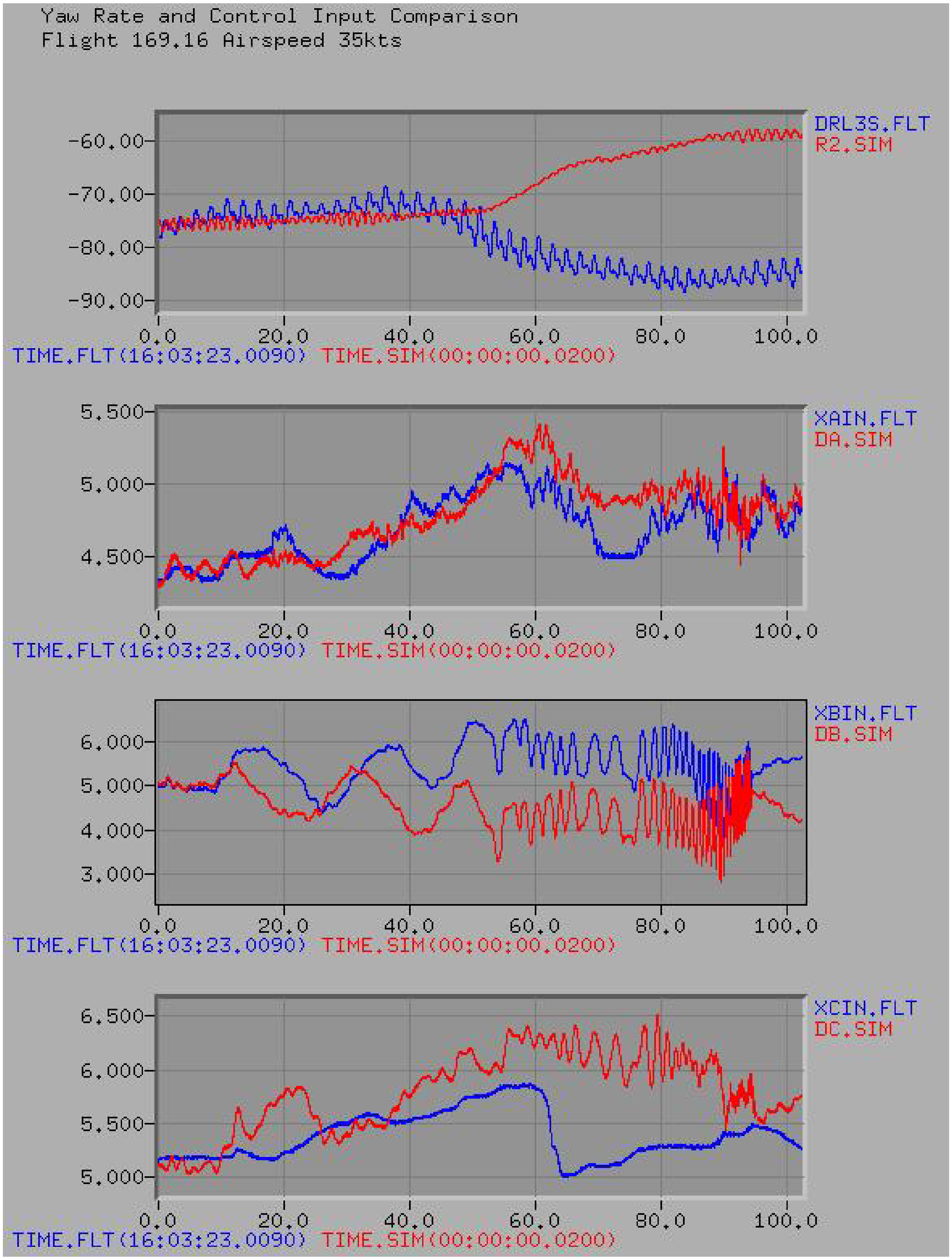
# APPENDIX D YAW DOF CORR PLOTS

Plots of yaw rates and control inputs for GenHel / Slung Load simulation with correcting equations. Both swivel and non-swivel cases are plotted for validation. Plotted are flight test data in blue versus simulation in red. Each plot window has four plots. The first plot is yaw rate in feet per second, the second is longitudinal stick deflection in inches, the third is lateral stick deflection in inches and the bottom plot is collective deflection in inches. Appendix D by Ehlers, G.


Yaw Rate and Control Input Comparison
Flight 172.17 Airspeed 20kts
DRL3S.FLT
R2.SIM
50.00
0.00
-50.00
0.0
20.0
40.0
60.0
80.0
100.0
TIME.FLT(16:41:36.2510) TIME.SIM(00:00:00.0200)
XAIN.FLT
DA.SIM
5.000
4.500
4.000
0.0
20.0
40.0
60.0
80.0
100.0
TIME.FLT(16:41:36.2510) TIME.SIM(00:00:00.0200)
XBIN.FLT
DB.SIM
7.000
6.000
5.000
4.000
3.000
0.0
20.0
40.0
60.0
80.0
100.0
TIME.FLT(16:41:36.2510) TIME.SIM(00:00:00.0200)
XCIN.FLT
DC.SIM
6.000
5.000
4.000
0.0
20.0
40.0
60.0
80.0
100.0
TIME.FLT(16:41:36.2510) TIME.SIM(00:00:00.0200)


Yaw Motions and Control Input Comparison
Flight 172.19 Airspeed 42kts
DRL3S.FLT
R2.SIM
XAIN.FLT
DA.SIM
XBIN.FLT
DB.SIM
XCIN.FLT
DC.SIM
TIME.FLT(16:46:48.4390) TIME.SIM(00:00:00.0200)


Yaw Motions and Control Input Comparison
Flight 172.46 Airspeed 58kts
DRL3S.FLT
R2.SIM
100.0
50.0
0.0
-50.0
-100.0
0.0
50.0
100.0
TIME.FLT(17:45:59.2700) TIME.SIM(00:00:00.0200)
XAIN.FLT
DA.SIM
6.000
5.000
4.000
0.0
50.0
100.0
TIME.FLT(17:45:59.2700) TIME.SIM(00:00:00.0200)
XBIN.FLT
DB.SIM
7.000
6.500
6.000
5.500
5.000
0.0
50.0
100.0
TIME.FLT(17:45:59.2700) TIME.SIM(00:00:00.0200)
XCIN.FLT
DC.SIM
6.000
5.500
5.000
4.500
4.000
3.500
0.0
50.0
100.0
TIME.FLT(17:45:59.2700) TIME.SIM(00:00:00.0200)


Yaw Rate and Control Input Comparison
Flight 169.16 Airspeed 35kts
DRL3S.FLT
R2.SIM
-60.00
-70.00
-80.00
-90.00
0.0
20.0
40.0
60.0
80.0
100.0
TIME.FLT(16:03:23.0090) TIME.SIM(00:00:00.0200)
XAIN.FLT
DA.SIM
5.500
5.000
4.500
XBIN.FLT
DB.SIM
6.000
5.000
4.000
3.000
XCIN.FLT
DC.SIM
6.500
6.000
5.500
5.000

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX E LOADSTAB.M

Matlab Program to check force and moment equations for an airfoil exposed to a relative wind from any angle. Program by Ehlers, G.

```
% Check program for control regime

clear
clc
d2r=pi/180;
%Flight Constants
Va=30;
rho=.0023768;

%Stall Transition Region
TH1=10;
TH2=30;

%Lift and Drag Coefficients
Cla=.1/d2r;
Cdo=.02;
Cdp=.5;

%Airfoil Dimentions entered [Horz,Vert]
Chord=[2,2];
Span=[5,3];
%Airfoil location
RH=[-5,0,-4];
RV=[-5,0,-5.5];

%Conversions
TH1=TH1*d2r;
TH2=TH2*d2r;
V=-Va*1.68781;

%Plots wanted 1=all, 2=airfoil-centered axis, 3=load-centered axis
pl=1;

n=51;
alf=linspace(-90,90,n)*d2r;
bet=linspace(-180,180,n)*d2r;

%Calculates first Horiz Stab then Vert Stab
for g = 1:2
   Ch=Chord(g);
   S = Chord(g)*Span(g);
   AR= Span(g)/Chord(g);
   e = .9;

   for i=1:length(alf)
      for j=1:length(bet)

         %Calculate u,v,w for full range of possibilities
```

```
         if g==1
            u = -V*cos(alf(i))*cos(bet(j));
            v = V*sin(bet(j));
            w = -V*sin(alf(i))*cos(bet(j));
         else
            u = -V*cos(alf(i))*cos(bet(j));
            v = -V*sin(alf(i))*cos(bet(j));
            w = -V*sin(bet(j));
         end

         %Calculate Theta and Beta for Stab
         TH=atan2(w,u);
         PS=atan2(v,u);

         %Relative Wind perp to span
         Vxy = sqrt(u^2+v^2);
         Vxz = sqrt(u^2+w^2);
         Vyz = sqrt(v^2+w^2);

         %Calculations for forces and moments

         %Prestall Region
         if (abs(TH) <= TH1) & (abs(PS) <= 1.571)
            F1(i,j) = -rho/2*Vxy*u*S*(Cdo+Cla^2*TH^2/(pi*e*AR));
            F2(i,j) = -rho/2*Vyz*v*S*Cdo;
            F3(i,j) = -rho/2*Vxz^2*S*Cla*TH;
            M1(i,j) = 0;
            M2(i,j) = -F3(i,j)*(Ch/4);
            M3(i,j) = 0;
         %Crossflow Region
         elseif (abs(TH) > TH2) | (abs(PS) > 1.571)
            F1(i,j) = -rho/2*Vxy*u*S*Cdo;
            F2(i,j) = -rho/2*Vyz*v*S*Cdo;
            F3(i,j) = -rho/2*Vxz*w*S*(Cdp*abs(sin(TH)));
            M1(i,j) = 0;
            M2(i,j) = 0;
            M3(i,j) = 0;
         %Stall Transition Region
         else
            u2   = Vxz*cos(TH2)*sign(cos(PS));
            Vxy2 = sqrt(u2^2+v^2);
            F1p1 = -rho/2*Vxy*u*S*(Cdo+Cla^2*TH1^2/(pi*e*AR));
            F2p1 = -rho/2*Vxy2*u2*S*Cdo;
            F1(i,j) = F1p1+(F2p1-F1p1)/(TH2-TH1)*(abs(TH)-TH1);
            M1(i,j) = 0;
            F2(i,j) = -rho/2*Vyz*v*S*Cdo;
            M3(i,j) = 0;
            w2   = Vxz*sin(TH2)*sign(sin(TH));
            Vxz2 = sqrt(u^2+w2^2);
            F1p2 = -rho/2*Vxz^2*S*(Cla*TH1*sign(TH));
            F2p2 = -
rho/2*Vxz2*w2*S*(Cdo*abs(cos(TH2))+Cdp*abs(sin(TH2)));
            F3(i,j) = F1p2+(F2p2-F1p2)/(TH2-TH1)*(abs(TH)-TH1);
            M2(i,j) = -F1p2*(Ch/4)+(F1p2*(Ch/4))/(TH2-TH1)*(abs(TH)-
TH1);
         end
      end
```

```
    end

    %Calculation of load body centered forces and moments
    if g==1;

       %Horiz Stab
       F1LH = F1;
       F2LH = F2;
       F3LH = F3;
       M1LH = -F2.*RH(3)+F3.*RH(2);
       M2LH = F1.*RH(3)-F3.*RH(1)+M2;
       M3LH = -F1.*RH(2)+F2.*RH(1);
    else
       %Vertical Stab
       F1LV = F1;
       F2LV = -F3;
       F3LV = F2;
       M1LV = F2.*RV(2)+F3.*RV(3);
       M2LV = F1.*RV(3)-F2.*RV(1);
       M3LV = -F1.*RV(2)-F3.*RV(1)+M2;
    end
 end
 %Total Forces load body centered
 F1L = F1LH+F1LV;
 F2L = F2LH+F2LV;
 F3L = F3LH+F3LV;
 M1L = M1LH+M1LV;
 M2L = M2LH+M2LV;
 M3L = M3LH+M3LV;

 if pl==1 | pl==2
    %3-D Graphs for forces
    figure
    colormap([0,0,0])
    mesh (bet/d2r,alf/d2r,F1)
    title ('Axial Forces body centered axis')
    ylabel ('Pitch Angle')
    xlabel ('Yaw Angle')
    grid on

    figure
    colormap([0,0,0])
    mesh (bet/d2r,alf/d2r,F2)
    title ('Lateral Forces body centered axis')
    ylabel ('Pitch Angle')
    xlabel ('Yaw Angle')
    grid on

    figure
    colormap([0,0,0])
    mesh (bet/d2r,alf/d2r,F3)
    title ('Vertical Forces body centered axis')
    ylabel ('Pitch Angle')
    xlabel ('Yaw Angle')
    grid on

    %3-D Graphs for moments
```

```
   figure
   colormap([0,0,0])
   mesh (bet/d2r,alf/d2r,M2)
   title ('Pitch Moments, body centered axis')
   ylabel ('Pitch Angle')
   xlabel ('Yaw Angle')
   grid on

   nn=[1,14,26,39,51];
   ymin=min([min(min(F1)),min(min(F2)),min(min(F3))]);
   ymax=max([max(max(F1)),max(max(F2)),max(max(F3))]);

   for ii=1:length(nn)
      %2-D Graphs

      %Force Plots
      figure
      subplot (4,2,1), plot(alf/d2r,F1(:,nn(ii)))
      title (['Axial Force vs Alpha, Beta =
',num2str(bet(nn(ii))/d2r)])
      axis([alf(1)/d2r,alf(n)/d2r,ymin,ymax])
      grid on

      subplot (4,2,3), plot(alf/d2r,F2(:,nn(ii)))
      title (['Lateral Force vs Alpha, Beta =
',num2str(bet(nn(ii))/d2r)])
      axis([alf(1)/d2r,alf(n)/d2r,ymin,ymax])
      ylabel ('Force')
         grid on

      subplot (4,2,5), plot(alf/d2r,F3(:,nn(ii)))
      title (['Normal Force vs Alpha, Beta =
',num2str(bet(nn(ii))/d2r)])
      axis([alf(1)/d2r,alf(n)/d2r,ymin,ymax])
      grid on

      subplot (4,2,2), plot(bet/d2r,F1(nn(ii),:))
      title (['Axial Force vs Beta, Alpha= ',num2str(alf(nn(ii))/d2r)])
      axis([bet(1)/d2r,bet(n)/d2r,ymin,ymax])
      grid on

      subplot (4,2,4), plot(bet/d2r,F2(nn(ii),:))
      title (['Lateral Force vs Beta, Alpha=
',num2str(alf(nn(ii))/d2r)])
      ylabel ('Force')
      axis([bet(1)/d2r,bet(n)/d2r,ymin,ymax])
      grid on

      subplot (4,2,6), plot(bet/d2r,F3(nn(ii),:))
      title (['Normal Force vs Beta, Alpha=
',num2str(alf(nn(ii))/d2r)])
      axis([bet(1)/d2r,bet(n)/d2r,ymin,ymax])
      grid on

      %Moment Plots
      subplot (4,2,7), plot(alf/d2r,M2(:,nn(ii)))
```

```
      title (['Pitching Moment vs Alpha, Beta =
',num2str(bet(nn(ii))/d2r)])
      axis([alf(1)/d2r,alf(n)/d2r,min(min(M2)),max(max(M2))])
      xlabel ('Alpha')
      grid on

      subplot (4,2,8), plot(bet/d2r,M2(:,nn(ii)))
      title (['Pitching Moment vs Beta, Alpha=
',num2str(alf(nn(ii))/d2r)])
      axis([bet(1)/d2r,bet(n)/d2r,min(min(M2)),max(max(M2))])
      xlabel ('Beta')
      grid on
   print

   end

elseif pl==1 | pl==3
   %Graphs for Load Body Centered Axis

   %Axial Forces
   figure
   colormap([0,0,0])
   mesh (bet/d2r,alf/d2r,F1LH)
   title ('Axial Forces Horz Stab, Load-body centerd axis')
   ylabel ('Pitch Angle')
   xlabel ('Yaw Angle')
   grid on

   figure
   colormap([0,0,0])
   mesh (bet/d2r,alf/d2r,F1LV)
   title ('Axial Forces Vert Stab, Load-body centerd axis')
   ylabel ('Pitch Angle')
   xlabel ('Yaw Angle')
   grid on

   figure
   colormap([0,0,0])
   mesh (bet/d2r,alf/d2r,F1L)
   title ('Total Axial Forces, Load-body centerd axis')
   ylabel ('Pitch Angle')
   xlabel ('Yaw Angle')
   grid on

   %Lateral Forces
   figure
   colormap([0,0,0])
   mesh (bet/d2r,alf/d2r,F2LH)
   title ('Lateral Forces Horz Stab, Load-body centerd axis')
      ylabel ('Pitch Angle')
      xlabel ('Yaw Angle')
   grid on

   figure
   colormap([0,0,0])
      mesh (bet/d2r,alf/d2r,F2LV)
      title ('Lateral Forces Vert Stab, Load-body centerd axis')
```

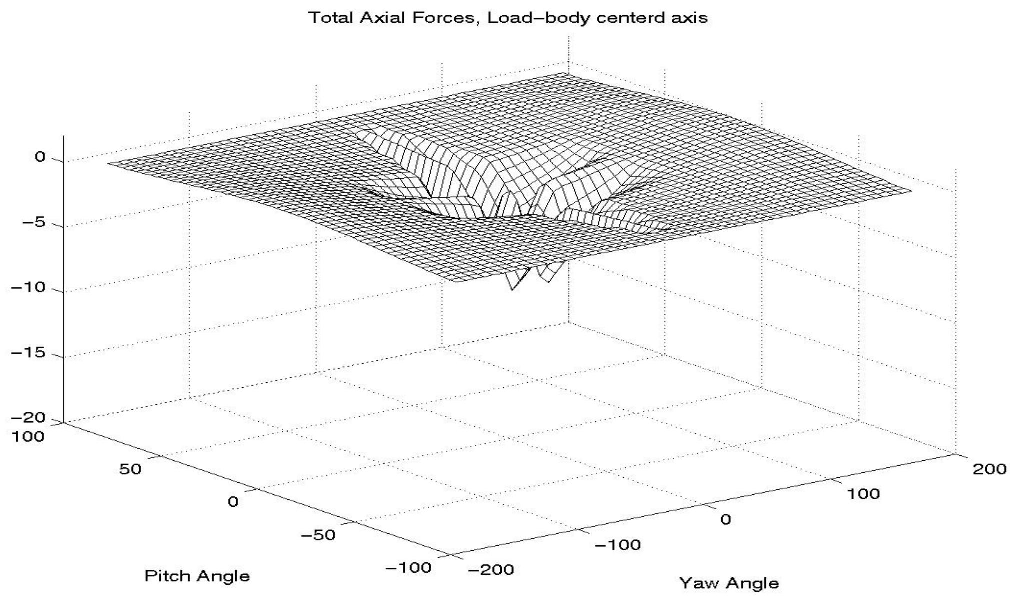
```
   ylabel ('Pitch Angle')
   xlabel ('Yaw Angle')
   grid on

   figure
   colormap([0,0,0])
   mesh (bet/d2r,alf/d2r,F2L)
   title ('Total Lateral Forces, Load-body centerd axis')
   ylabel ('Pitch Angle')
   xlabel ('Yaw Angle')
grid on

   %Vertical Forces
   figure
colormap([0,0,0])
   mesh (bet/d2r,alf/d2r,F3LH)
   title ('Vertical Forces Horz Stab, Load-body centerd axis')
   ylabel ('Pitch Angle')
   xlabel ('Yaw Angle')
grid on

   figure
   colormap([0,0,0])
   mesh (bet/d2r,alf/d2r,F3LV)
   title ('Vertical Forces Vert Stab, Load-body centerd axis')
   ylabel ('Pitch Angle')
   xlabel ('Yaw Angle')
grid on

figure
colormap([0,0,0])
   mesh (bet/d2r,alf/d2r,F3L)
   title ('Total Vertical Forces, Load-body centerd axis')
   ylabel ('Pitch Angle')
   xlabel ('Yaw Angle')
grid on

   %Roll Moment
   figure
   colormap([0,0,0])
   mesh (bet/d2r,alf/d2r,M1LH)
   title ('Roll Moment Horz Stab, Load-body centerd axis')
   ylabel ('Pitch Angle')
   xlabel ('Yaw Angle')
grid on

   figure
colormap([0,0,0])
   mesh (bet/d2r,alf/d2r,M1LV)
   title ('Roll Moment Vert Stab, Load-body centerd axis')
   ylabel ('Pitch Angle')
   xlabel ('Yaw Angle')
grid on

   figure
   colormap([0,0,0])
   mesh (bet/d2r,alf/d2r,M1L)
```

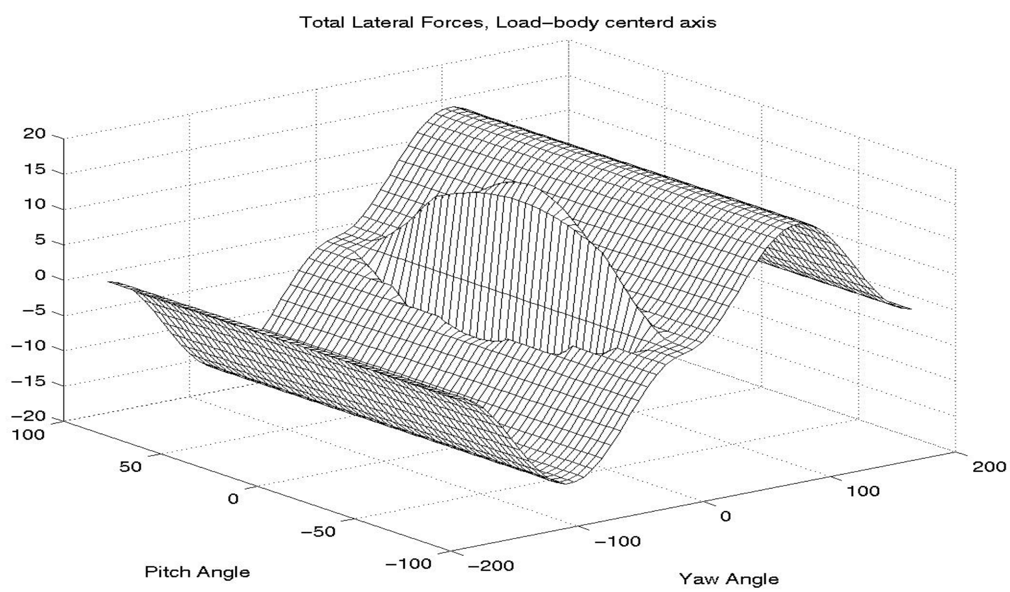
```
   title ('Total Roll Moment, Load-body centerd axis')
   ylabel ('Pitch Angle')
   xlabel ('Yaw Angle')
grid on

   %Pitch Moment
   figure
   colormap([0,0,0])
   mesh (bet/d2r,alf/d2r,M2LH)
   title ('Pitch  Moment Horz Stab, Load-body centerd axis')
   ylabel ('Pitch Angle')
   xlabel ('Yaw Angle')
grid on

   figure
colormap([0,0,0])
   mesh (bet/d2r,alf/d2r,M2LV)
   title ('Pitch Moment Vert Stab, Load-body centerd axis')
   ylabel ('Pitch Angle')
   xlabel ('Yaw Angle')
grid on

   figure
   colormap([0,0,0])
   mesh (bet/d2r,alf/d2r,M2L)
   title ('Total Pitch Moment, Load-body centerd axis')
   ylabel ('Pitch Angle')
   xlabel ('Yaw Angle')
grid on

   %Yaw Moment
   figure
   colormap([0,0,0])
   mesh (bet/d2r,alf/d2r,M3LH)
   title ('Yaw Moment Horz Stab, Load-body centerd axis')
   ylabel ('Pitch Angle')
   xlabel ('Yaw Angle')
grid on

   figure
colormap([0,0,0])
   mesh (bet/d2r,alf/d2r,M3LV)
   title ('Yaw Moment Vert Stab, Load-body centerd axis')
   ylabel ('Pitch Angle')
   xlabel ('Yaw Angle')
grid on

   figure
   colormap([0,0,0])
   mesh (bet/d2r,alf/d2r,M3L)
   title ('Total Yaw Moment, Load-body centerd axis')
   ylabel ('Pitch Angle')
   xlabel ('Yaw Angle')
grid on

nn=[1,14,26,39,51];
for ii=1:length(nn)
```

```
      %2-D Moment Graphs for load-body centered axis
      figure
      subplot (3,2,1), plot(alf/d2r,M2L(:,nn(ii)))
      title (['Pitch Moment vs Alpha, Beta =
',num2str(bet(nn(ii))/d2r)])
      axis([alf(1)/d2r,alf(n)/d2r,-100,100])
      grid on

      subplot (3,2,3), plot(alf/d2r,M1L(:,nn(ii)))
      title (['Rolling Moment vs Alpha, Beta =
',num2str(bet(nn(ii))/d2r)])
      axis([alf(1)/d2r,alf(n)/d2r,-100,100])
      ylabel ('Moment')
      grid on

      subplot (3,2,5), plot(alf/d2r,M3L(:,nn(ii)))
      title (['Yaw Moment vs Alpha, Beta = ',num2str(bet(nn(ii))/d2r)])
      axis([alf(1)/d2r,alf(n)/d2r,-100,100])
      xlabel ('Alpha')
      grid on

      subplot (3,2,2), plot(bet/d2r,M2L(nn(ii),:))
      title (['Pitching Moment vs Beta, Alpha=
',num2str(alf(nn(ii))/d2r)])
      axis([bet(1)/d2r,bet(n)/d2r,-100,100])
      grid on

      subplot (3,2,4), plot(bet/d2r,M1L(nn(ii),:))
      title (['Rolling Moment vs Beta, Alpha=
',num2str(alf(nn(ii))/d2r)])
      ylabel ('Moment')
      axis([bet(1)/d2r,bet(n)/d2r,-100,100])
      grid on

      subplot (3,2,6), plot(bet/d2r,M3L(nn(ii),:))
      title (['Yaw Moment vs Beta, Alpha= ',num2str(alf(nn(ii))/d2r)])
      axis([bet(1)/d2r,bet(n)/d2r,-100,100])
      xlabel ('Beta')
      grid on
   end
end
```

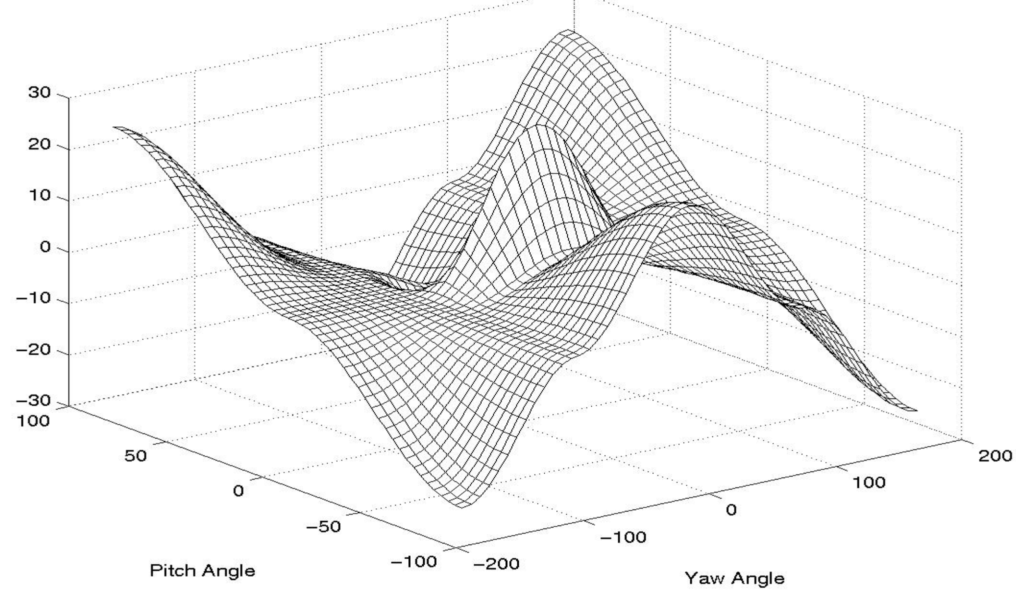
## APPENDIX F  STABILIZER FORCES PLOTS

Plots of forces and moments for an stabilization device  exposed to a relative wind from any angle.  Forces are in $lbs_f$, moments are in ft-lbs and angles are in degrees. Appendix F by Ehlers, G.

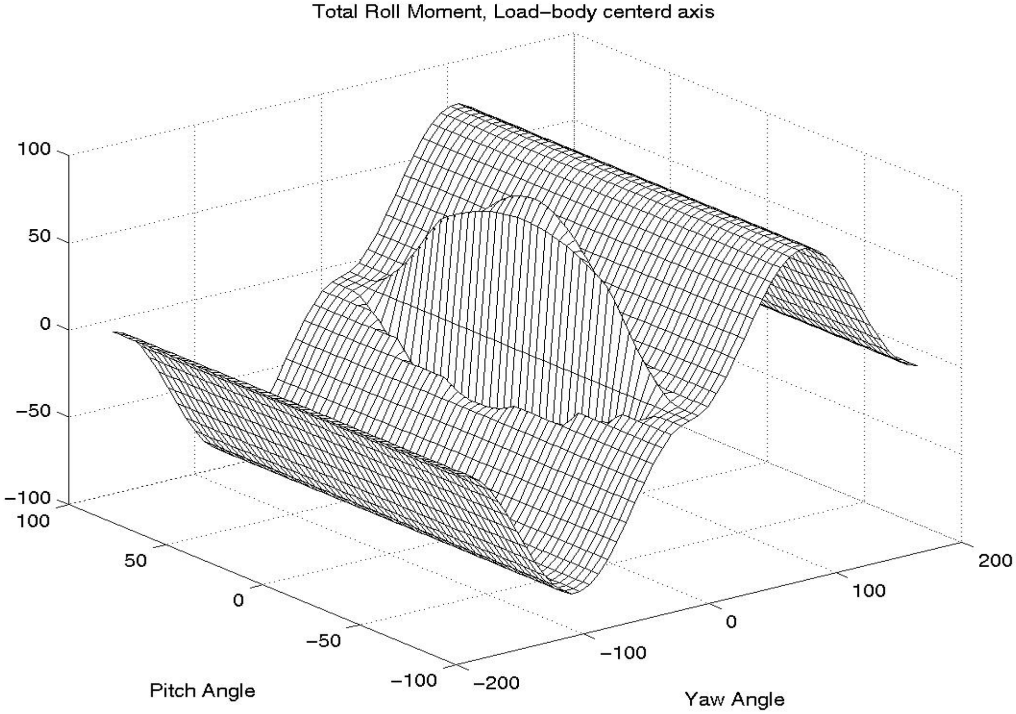


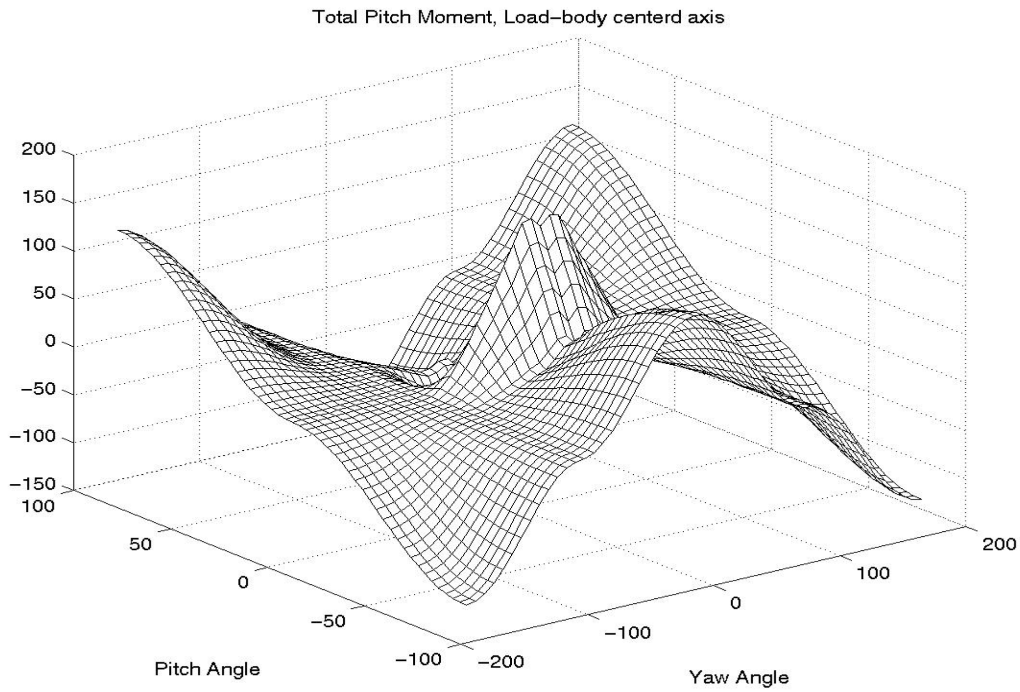



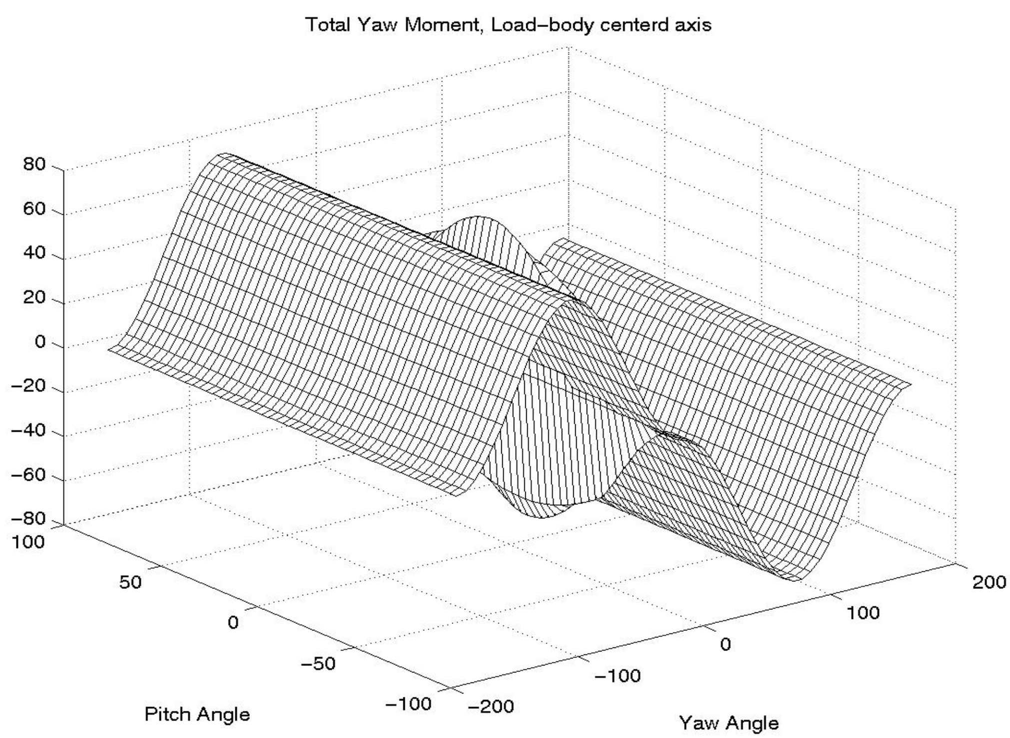
Total Vertical Forces, Load–body centerd axis

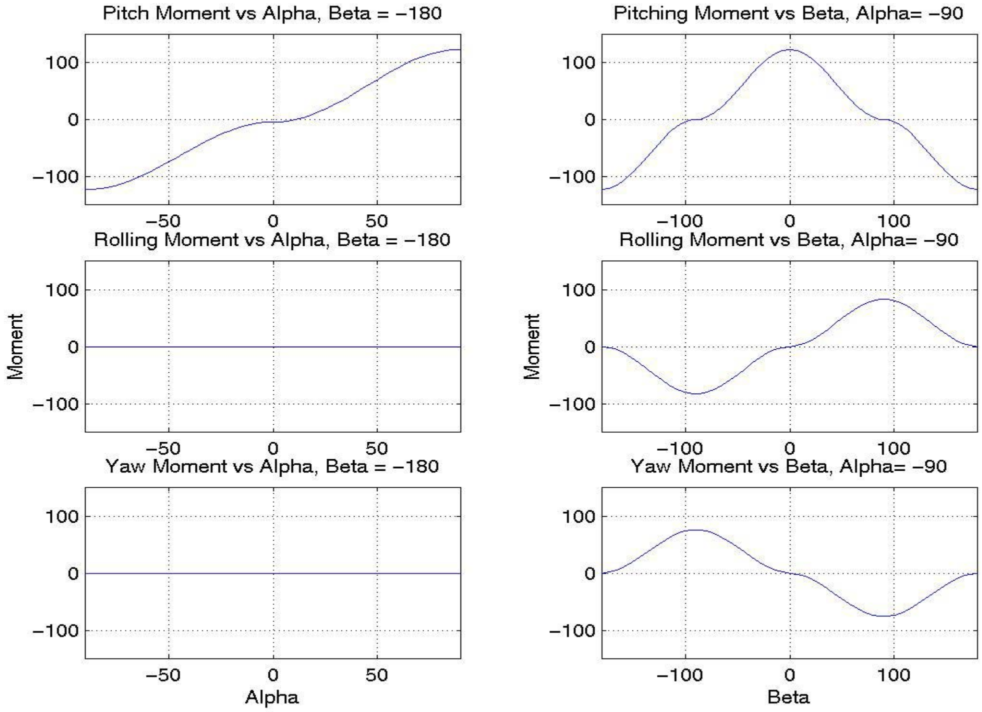


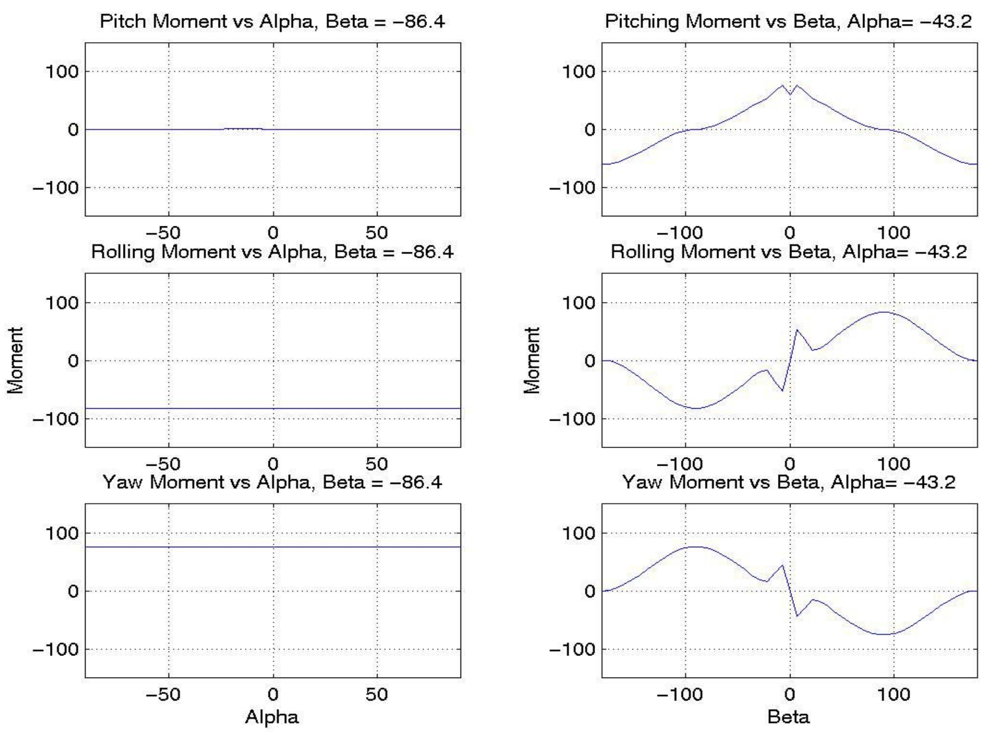

Total Roll Moment, Load–body centerd axis

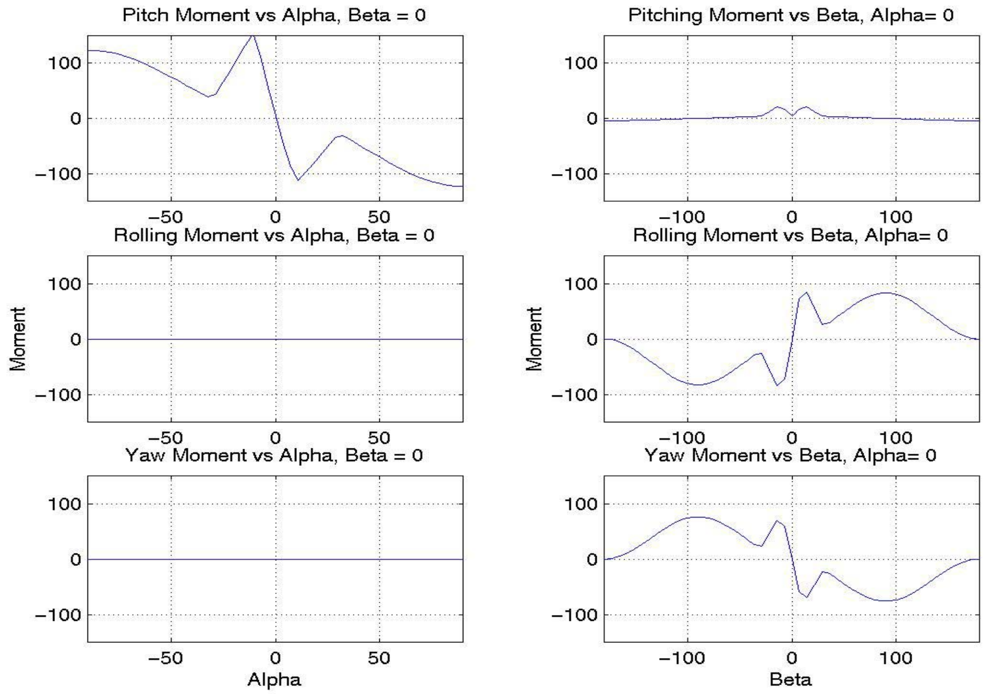


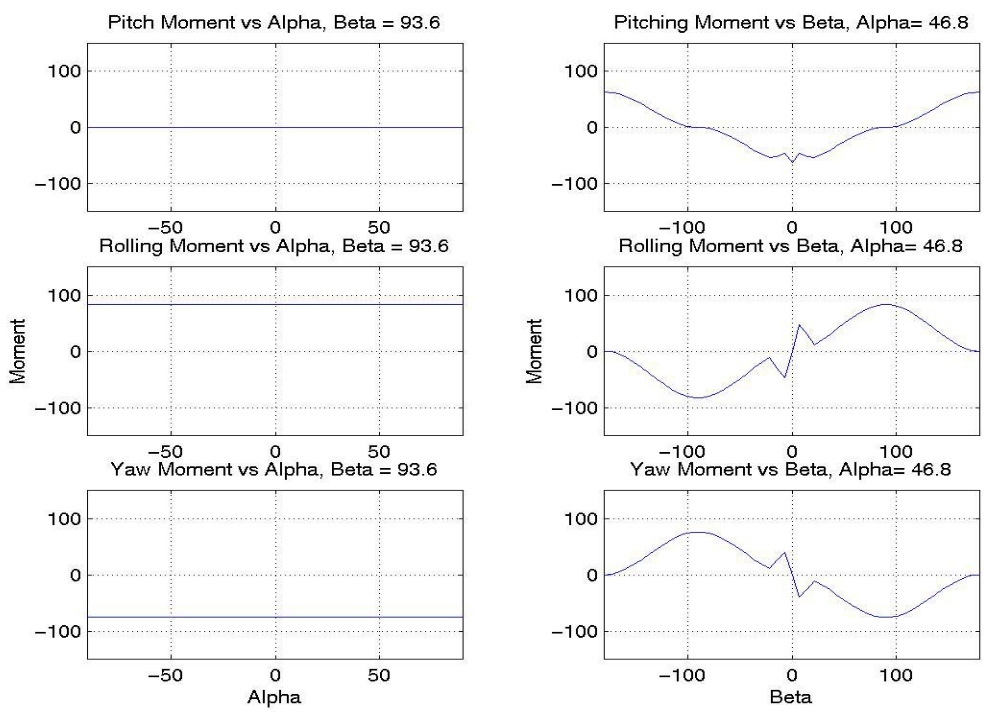

Total Pitch Moment, Load-body centerd axis
200
150
100
50
0
−50
−100
−150
100
50
0
−50
−100
Pitch Angle
−200
−100
0
100
200
Yaw Angle

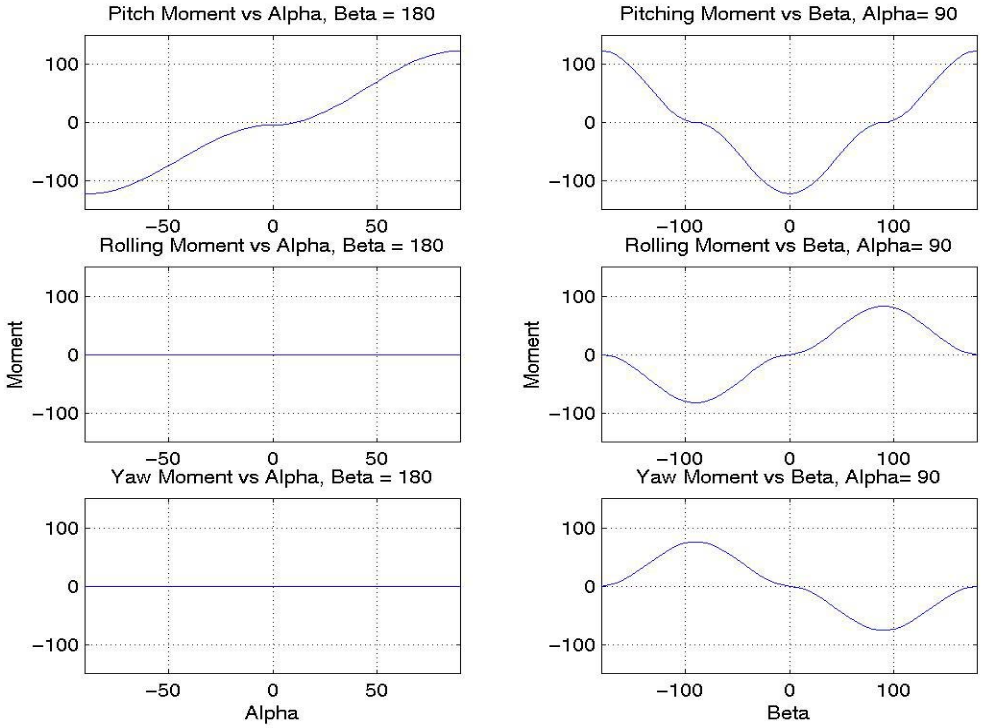

Total Yaw Moment, Load-body centerd axis
80
60
40
20
0
−20
−40
−60
−80
100
50
0
−50
−100
Pitch Angle
−200
−100
0
100
200
Yaw Angle


Pitch Moment vs Alpha, Beta = −180
Pitching Moment vs Beta, Alpha= −90
Rolling Moment vs Alpha, Beta = −180
Rolling Moment vs Beta, Alpha= −90
Yaw Moment vs Alpha, Beta = −180
Yaw Moment vs Beta, Alpha= −90
Moment
Alpha
Beta


Pitch Moment vs Alpha, Beta = −86.4
Pitching Moment vs Beta, Alpha= −43.2
Rolling Moment vs Alpha, Beta = −86.4
Rolling Moment vs Beta, Alpha= −43.2
Yaw Moment vs Alpha, Beta = −86.4
Yaw Moment vs Beta, Alpha= −43.2
Moment
Alpha
Beta


Pitch Moment vs Alpha, Beta = 0
Pitching Moment vs Beta, Alpha= 0
Rolling Moment vs Alpha, Beta = 0
Rolling Moment vs Beta, Alpha= 0
Yaw Moment vs Alpha, Beta = 0
Yaw Moment vs Beta, Alpha= 0
Moment
Alpha
Beta


Pitch Moment vs Alpha, Beta = 93.6
Pitching Moment vs Beta, Alpha= 46.8
Rolling Moment vs Alpha, Beta = 93.6
Rolling Moment vs Beta, Alpha= 46.8
Yaw Moment vs Alpha, Beta = 93.6
Yaw Moment vs Beta, Alpha= 46.8
Moment
Alpha
Beta


Pitch Moment vs Alpha, Beta = 180
Pitching Moment vs Beta, Alpha= 90
Rolling Moment vs Alpha, Beta = 180
Rolling Moment vs Beta, Alpha= 90
Yaw Moment vs Alpha, Beta = 180
Yaw Moment vs Beta, Alpha= 90
Moment
Alpha
Beta
100
0
−100
−50
50

# APPENDIX G LOADSTAB.F

Stabilization Module for GenHel / Slung Load simulation. Program by Ehlers, G.

```
C  File GenHel/batch/sl/loadstab.f.........................05 June 01
George Ehlers
C Fin tale  stabilizer for slung loads

      SUBROUTINE LOADSTAB

      INCLUDE 'slvars.cmn'

      REAL
     X  MACH, M1H, M2H, M3H, M1V, M2V, M3V, MXF, MYF, MZF

      EQUIVALENCE (OM22(1),  P2)
      EQUIVALENCE (OM22(2),  Q2)
      EQUIVALENCE (OM22(3),  R2)

      DATA E, PI/.9, 3.1415927/


C  Calcualtion of stab perameters
      AH  = CH*SH
      AV  = CV*SV
      ARH = SH/CH
      ARV = SV/CV

C  Wind Parameters, UU,VV,WW used instead of U,V,W to prevent naming
errors

      UU = va2s2(1) + q2*rv(3) - r2*rv(2)
      VV = va2s2(2) + r2*rv(1) - p2*rv(3)
      WW = va2s2(3) + p2*rv(2) - q2*rv(1)

C  Relative Wind calculations
       Vxy = SQRT(UU**2+VV**2)
       Vxz = SQRT(UU**2+WW**2)
       Vyz = SQRT(VV**2+WW**2)

C  Calculation of active stab gains
      IF (ACTIVE .NE. 1) THEN
         DEFH = 0
         DEFV = 0
      ELSE
         DEFH = -GAINH*Q2
         IF (DEFH .GT. .349) THEN
            DEFH = -.349*SIGN(Q2)
         ENDIF
         DEFV = -GAINV*R2
         IF (DEFV .GT. .349) THEN
            DEFV = -.349*SIGN(R2)
         ENDIF
      ENDIF
```

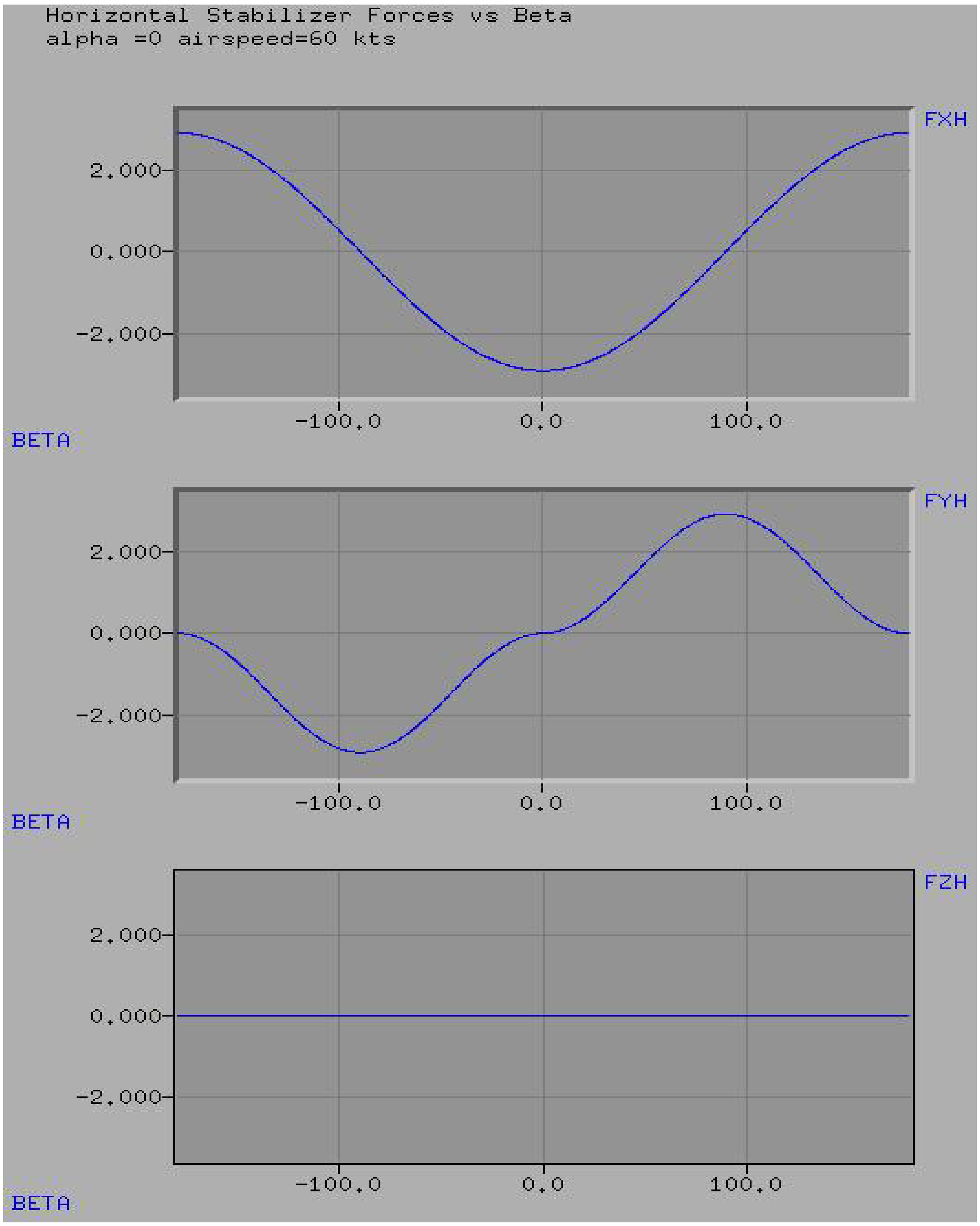
```
C  Calculations for forces and moments

C  Horizontal Stab

C  Calculation of angle of attack and yaw
      TH = atan2(WW,UU)
      PS = atan2(-VV,UU)

C  Calculation of 3D lift coeffic and conversions
      CLA  = (CLAH*r2d)*ARH/(2+sqrt(4+ARH**2))
      CLD   = CLDH
      CDO   = CDOH
      CDP   = CDPH
      STAL1 = STALH1/R2D
      STAL2 = STALH2/R2D

C  Prestall Region
      IF (ABS(TH) .LE. STAL1 .AND. ABS(PS) .LE. 1.571) THEN
         F1H = -RHO/2*Vxy*UU*AH*(Cdo+Cla**2*TH**2/(PI*E*ARH))
         F2H = -RHO/2*Vyz*VV*AH*Cdo
         F3H = -RHO/2*Vxz**2*AH*(Cla*TH+Cld*DEFH)
         M1H = 0
         M2H = F3H*(CH/4)
         M3H = 0
C  Crossflow Region
      ELSEIF (ABS(TH) .GT. STAL2 .OR. ABS(PS) .GT. 1.571)
     X THEN
         F1H = -RHO/2*Vxy*UU*AH*Cdo
         F2H = -RHO/2*Vyz*VV*AH*Cdo
         F3H = -RHO/2*Vxz*WW*AH*(Cdp*ABS(SIN(TH)))
         M1H = 0
         M2H = 0
         M3H = 0
C  Stall Transition Region
      ELSE
         UU2   = Vxz*COS(STAL2)*SIGN(1,COS(PS))
         Vxy2  = SQRT(UU2**2+VV**2)
         F1p1  = -RHO/2*Vxy*UU*AH*(Cdo+Cla**2*STAL1**2/
     X          (PI*E*ARH))
         F2p1  = -RHO/2*Vxy2*UU2*AH*Cdo
         F1H   = F1p1+(F2p1-F1p1)/(STAL2-STAL1)*(abs(TH)-
     X          STAL1)
         F2H   = -RHO/2*Vyz*VV*AH*Cdo
         WW2   = Vxz*SIN(STAL2)*SIGN(1,SIN(TH))
         Vxz2  = SQRT(UU**2+WW2**2)
         F1p2  = -RHO/2*Vxz**2*AH*(Cla*STAL1*sign(1,TH)+Cld
     X          *DEFH)
         F2p2  = -RHO/2*Vxz2*WW*AH*(Cdo*ABS(COS(STAL2))
     X           +Cdp*ABS(SIN(STAL2)))
         F3H   = F1p2+(F2p2-F1p2)/(STAL2-STAL1)*(abs(TH)
     X          -STAL1)
         M1H   = 0
         M2H   = F1p2*(CH/4)+(F1p2*(CH/4))/(STAL2-STAL1)
     X          *(ABS(TH)-STAL1)
         M3H   = 0
      ENDIF
```

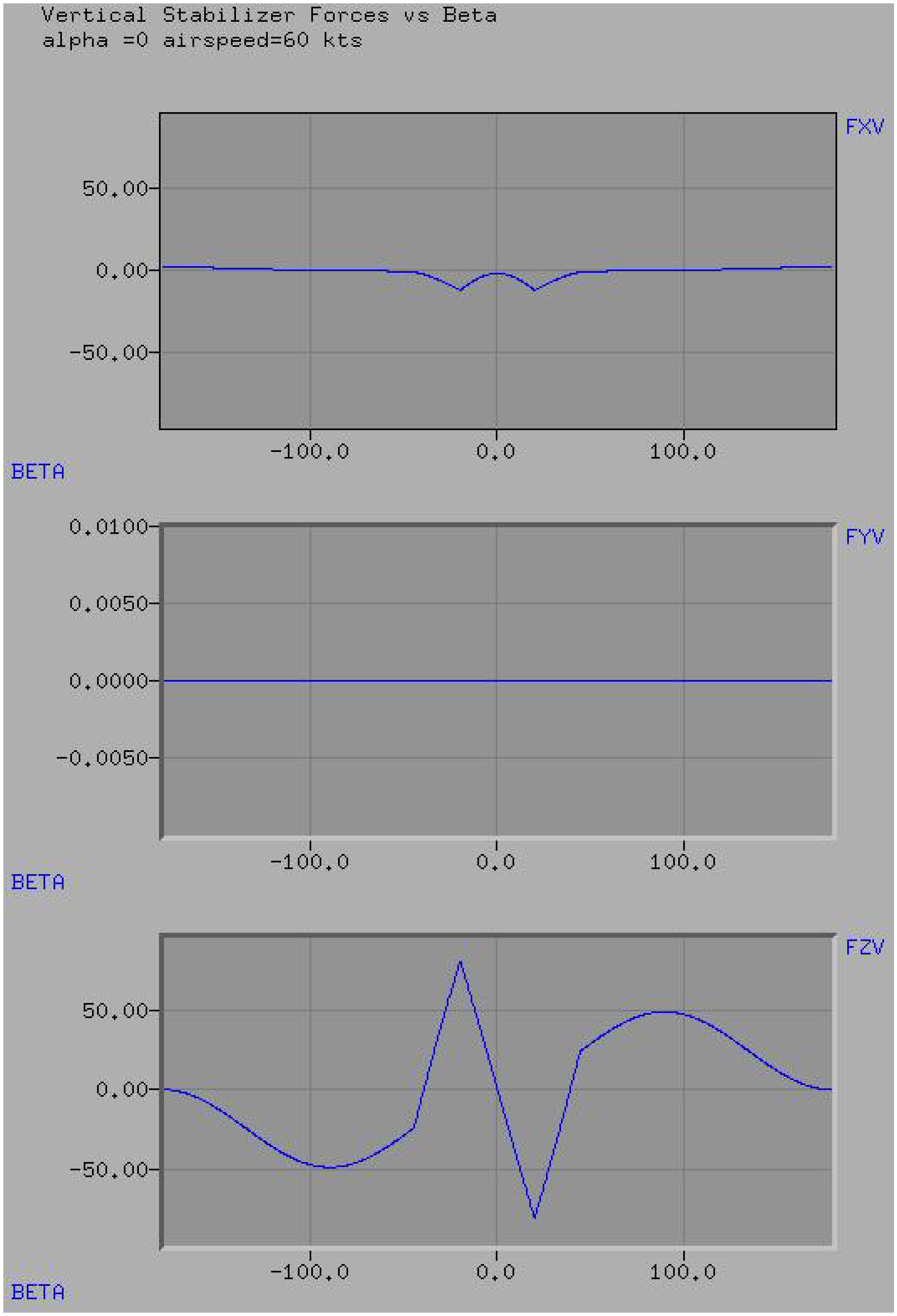
```
C  Vertical Stab

C  Calculation of 3D lift coeffic and conversions
      CLA  = (CLAV*r2d)*ARV/(2+sqrt(4+ARV**2))
      CLD   = CLDV
      CDO   = CDOV
      CDP   = CDPV
      STAL1 = STALV1/R2D
      STAL2 = STALV2/R2D
C  Transformation of wind components for Vertical Stab
         VV2=VV
         VV=-WW
         WW=VV2

         TH = ATAN2(WW,UU)
         PS = ATAN2(-VV,UU)

         Vxy = SQRT(UU**2+VV**2)
         Vxz = SQRT(UU**2+WW**2)
         Vyz = SQRT(VV**2+WW**2)

C  Prestall Region
      IF (ABS(TH) .LE. STAL1 .AND. ABS(PS) .LE. 1.571) THEN
         F1V = -RHO/2*Vxy*UU*AV*(CDO+Cla**2*TH**2/(PI*E*ARV))
         F2V = -RHO/2*Vyz*VV*AV*CDO
         F3V = -RHO/2*Vxz**2*AV*(Cla*TH+Cld*DEFV)
         M1V = 0
         M2V = F3V*(CV/4)
         M3V = 0
C  Crossflow Region
      ELSEIF (ABS(TH) .GT. STAL2 .OR. ABS(PS) .GT. 1.571)
     X            THEN
         F1V = -RHO/2*Vxy*UU*AV*CDO
         F2V = -RHO/2*VYZ*VV*AV*CDO
         F3V = -RHO/2*Vxz*WW*AV*(Cdp*ABS(SIN(TH)))
         M1V = 0
         M2V = 0
         M3V = 0
C  Stall Transition Region
      ELSE
         UU2   = Vxz*COS(STAL2)*SIGN(1,COS(TH))
         Vxy2  = SQRT(UU2**2+VV**2)
         F1p1  = -RHO/2*Vxy*UU*AV*(CDO+Cla**2*STAL1**2/
     X          (PI*E*ARV))
         F2p1  = -RHO/2*Vxy2*UU2*AV*CDO
         F1V   = F1p1+(F2p1-F1p1)/(STAL2-STAL1)*(abs(TH)-
     X          STAL1)
         F2V   = -RHO/2*Vyz*VV*AV*CDO
         WW2   = Vxz*SIN(STAL2)*SIGN(1,SIN(TH))
         Vxz2  = SQRT(UU**2+WW2**2)
         F1p2  = -RHO/2*Vxz**2*AV*(Cla*STAL1*sign(1,TH)+Cld*
     X          DEFV)
         F2p2  = -RHO/2*Vxz2*WW*AV*(CDO*ABS(COS(STAL2))
     X          +Cdp*ABS(SIN(STAL2)))
         F3V   = F1p2+(F2p2-F1p2)/(STAL2-STAL1)*(abs(TH)-
     X          STAL1)
```

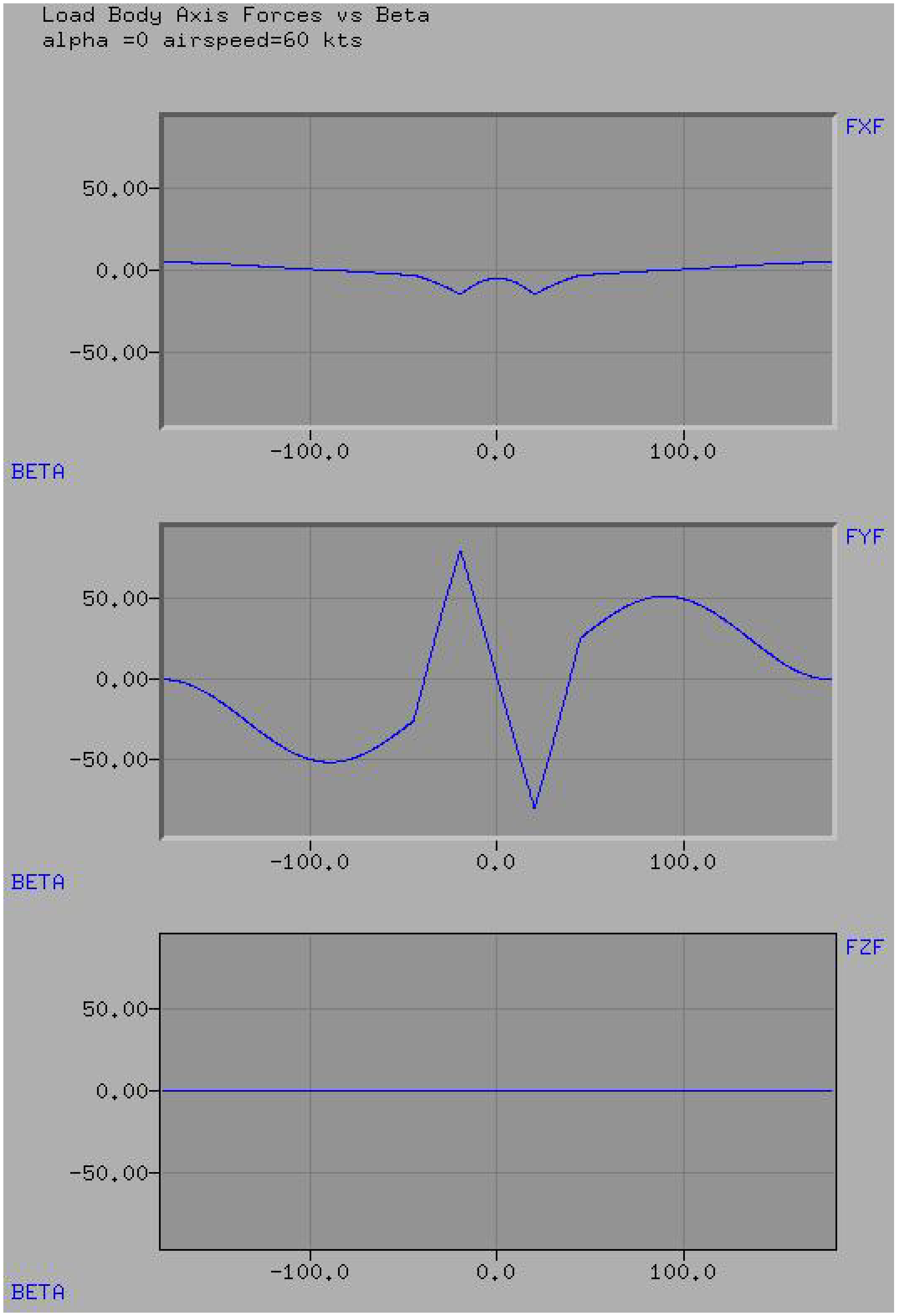
```
         M1V   = 0
         M2V   = -F1p2*(CV/4)+(F1p2*(CV/4))/(STAL2-STAL1)
     X            *(ABS(TH)-STAL1)
         M3V   = 0

      ENDIF

C  Calculations of forces and moments from stabs in load body centered
axis

      FXF = F1H+F1V
      FYF = F2H-F3V
      FZF = F3H+F2V
      MXF = -F2H*RH(3)+F3H*RH(2) - F2V*RV(2)-F3V*RV(3)
      MYF = F1H*RH(3)-F3H*RH(1)+M2H + F1V*RV(3)+F2V*RV(1)
      MZF = -F1H*RH(2)+F2H*RH(1) - F1V*RV(2)+F3V*RV(1)-M2V

      FA22(1) = FA22(1) + FXF
      FA22(2) = FA22(2) + FYF
      FA22(3) = FA22(3) + FZF
      MA22(1) = MA22(1) + MXF
      MA22(2) = MA22(2) + MYF
      MA22(3) = MA22(3) + MZF


      RETURN

      END
```

# APPENDIX H WIND AXIS STAB FORCES

Wind Axis forces for horizontal, vertical and total forces produced by the stabilization device. Plotted are the x, y and z forces (FX, FY, FZ)in $lbs_f$ versus yaw angle in degrees.




Vertical Stabilizer Forces vs Beta
alpha =0 airspeed=60 kts
FXV
50.00
0.00
-50.00
-100.0
0.0
100.0
BETA
FYV
0.0100
0.0050
0.0000
-0.0050
-100.0
0.0
100.0
BETA
FZV
50.00
0.00
-50.00
-100.0
0.0
100.0
BETA


Load Body Axis Forces vs Beta
alpha =0 airspeed=60 kts
FXF
50.00
0.00
-50.00
-100.0
0.0
100.0
BETA
FYF
50.00
0.00
-50.00
-100.0
0.0
100.0
BETA
FZF
50.00
0.00
-50.00
-100.0
0.0
100.0
BETA

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX I LOADSTAB.DAT

User interface for load stabilization module for GenHel / Slung Load simulation. All stabilization parameters are entered here to be read by loadstab.f. Program by Ehlers, G.

```
C  File GenHel/batch/loadstabil.dat.........................05 June 01
George Ehlers
C  Data input file for fin type stabilizer for slung loads
C  To be read by ghsl_init.f

C     ACTIVE - Passive / Active stabilization Flag 1=Active 0=Passive

C   Stabilizer Data, Final letter designates vertical or horizontal
stab
C     Gain - Active feedback gain
C     C    - Chord Length
C     S    - Airfoil Span
C     CLA  - 2-D Coefficent of lift of airfoil  section
C     CLD  - Coefficient of lift due to control deflection
C     CDO  - Zero Lift Coefficient of drag
C     CDP  - Profile Drag Coefficient
C     STAL1- Angle of Pitch Where Flow Seperation Begins for Horz Stab
C     STAL2- Angle of Pitch of Complete Airfoil Stall for Horz Stab
C     R    - X, Y and Z Distance of airfoil center to load CG

 &STABDAT
      ACTIVE = 5
      GAINH = 1
      CH    = 2
      SH    = 5
      CLAH  = .1
      CLDH  = .3
      CDOH  = .02
      CDPH  = .8
      STALH1= 20
      STALH2= 45
      RH(1) = -8
      RH(2) = 0
      RH(3) = -6

      GAINV = 1
      CV    = 2
      SV    = 4
      CLAV  = .1
      CLDV  = .3
      CDOV  = .02
      CDPV  = .8
      STALV1= 20
      STALV2= 45
      RV(1) = -8
      RV(2) = 0
      RV(3) = -8
```

```
&END
```

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX J GHSLMC.F

Slung Load module for multicable sling. Correction to the yaw degree of freedom and yaw resistance at the hook were make in this module. Program by Tyson, P. modified by Ehlers, G.

```
C  file GenHel/batch/sl/ghslmc.f, originally slmc.f ........ started 1
july 96
C  single lift, multi-cable suspension.
C  modified for use with GENHEL UH-60A model
C  mods jun 00 ... clean out auxilliary varbs computed for sldriver.
Add load acclr signals amgs2
C  Yaw DOF corrections made June 01 by George Ehlers

      SUBROUTINE GHSLMC

      INCLUDE 'slvars.cmn'

      REAL T1N(3,3), TN1(3,3), A22(3,3), A23(3,3), A22J1I(3,3),
     X     A23J2I(3,3), RJR1(3,3), RJR2(3,3),
     X     TV12(3), TV22(3), TV2N(3), TV3N(3), TV4N(3), TV5N(3),
     X     TV6N(3), TV7N(3), TV8N(3), CA11(3), DAU2N(3), JOM11(3),
     X     JOM22(3), X11(3), X22(3), FO1N(3), FO2N(3), MO11(3),
     X     MO22(3), HTDIFO(3), VSON(3), V1SN(3), V1S2SN(3), COR1N(3),
     X     COR2N(3), DRA2SN(3), RSON(3), FC12(3),
     X     SF1N(3), SF2N(3), SM11(3), SM22(3), SUMM11(3), SUMM22(3),
     X     DDRA2SN(3), DV1S2SN(3), DVSN(3), ADU2N(3), OSSFN(3),
     X     OISSFN(3), T1T2(3,3),DYNAM,SWIRL, RWIND ,DELPS, KR2, KPS2,
     X     KSWRL, KDYN, SLWND, KPS, YMHOOK, K21(3)
      REAL
     X     II1XX, II1YY, II1ZZ, II1XZ, II2XX, II2YY, II2ZZ, II2XZ,
     X     M1, M2, MU12, M1PM2, M2OM, NQSL,
     X     ODU(12), ODQ(12), DLCJ(8), RAJ2(3,8), KCJ2(3,8),
     X     DELQIC(12), C(4),
     X     K2N(3), KN2(3),
     X     RA2SN(3), DV2S2(3), DV2SS2(3)
      EQUIVALENCE (U(1)    , V1SN(1) )
      EQUIVALENCE (Q(6)    , PS1 )
      EQUIVALENCE (TN2(1,3), K2N(1) )
      EQUIVALENCE (T2N(1,3), KN2(1) )
      EQUIVALENCE (OM22(1),  P2)
      EQUIVALENCE (OM22(2),  Q2)
      EQUIVALENCE (OM22(3),  R2)
      EQUIVALENCE (DOM22(1), DP2)
      EQUIVALENCE (DOM22(2), DQ2)
      EQUIVALENCE (DOM22(3), DR2)

C     COMPUTE DEPENDENT PARAMETERS AND LOAD-SUSPENSION
C     POSITION COORDINATES FOR STATIC EQUILIBRIUM.

C--- idle mode (not used)
      IF (IMODE.EQ.0) RETURN
      DTO2 = .5*DT
```

```
C--- initial conditions
C  current helicopter mass and inertia from STRIKE
      M1      = W1/G
      TMP1    = I1XZ/(I1XX*I1ZZ)
      TMP2    = 1 - TMP1*I1XZ
      II1XX   = 1/(I1XX*TMP2)
      II1XZ   = TMP1/TMP2
      II1YY   = 1/I1YY
      II1ZZ   = 1/(I1ZZ*TMP2)

C  load parameters
      M2      = W2/G
      TMP1    = I2XZ/(I2XX*I2ZZ)
      TMP2    = 1 - TMP1*I2XZ
      II2XX   = 1/(I2XX*TMP2)
      II2XZ   = TMP1/TMP2
      II2YY   = 1/I2YY
      II2ZZ   = 1/(I2ZZ*TMP2)

C  compute derived mass parameters
      MU12    = 1/M1 + 1/M2
      M1PM2   = M1 + M2
      M2OM    = M2/M1PM2

C--- initial conditions (trim) mode
C  load-suspension states, add load yaw initial offset for yaw
stability tests
      IF (IMODE.LT.0) THEN
         CALL GHSLMC_IC
        PS2     = PS2 + DELPS2O/R2D
         DO 10 I = 7,12
         U(I)    = 0.
 10      DQ(I)   = 0.
         OM22(3) = R2OD/R2D
         DPS2    = OM22(3)*COS(PH2)/COS(TH2)

C  NS counts records stored
         NS    = 0
         DTO2  = 0.

C  magnetic dip angle at moffett
c         DIP   = 61.25/R2D
c         SDIP  = SIN(DIP)
c         CDIP  = COS(DIP)
      END IF

C************* OPERATE CODE ****************

C***  SEC 100: read in HC states and position kinematics

C  read in HC states from STRIKE to q, dq, u, T1N
      PH1      =  A(  4)
      TH1      =  A(  5)
      PS1      =  A(  6)
      R1SN(1)  =  A(106)
      R1SN(2)  =  A(107)
```

```
      R1SN(3)  = -A(176)
      DO 102 I = 1,3
      V1SN(I)  =  A(63 + I)
      DQ(I)    = V1SN(I)
      OM11(I)  =  A(36 + I)
 102  DQ(3+I)  =  A( 6 + I)
      DO 101 I = 1,3
      DO 101 J = 1,3
      T1N(I,J) =  A(15+I+(J-1)*3)
 101  TN1(J,I) =  A(15+I+(J-1)*3)

C  position kinematics
      SPH1     = SIN(PH1)
      CPH1     = COS(PH1)
      STH1     = SIN(TH1)
      CTH1     = COS(TH1)
      SPH2     = SIN(PH2)
      CPH2     = COS(PH2)
      STH2     = SIN(TH2)
      CTH2     = COS(TH2)
      SPS2     = SIN(PS2)
      CPS2     = COS(PS2)
      T2N(1,1) = CPS2*CTH2
      T2N(1,2) = CTH2*SPS2
      T2N(1,3) = -STH2
      T2N(2,1) = CPS2*SPH2*STH2-CPH2*SPS2
      T2N(2,2) = SPH2*SPS2*STH2+CPH2*CPS2
      T2N(2,3) = CTH2*SPH2
      T2N(3,1) = CPH2*CPS2*STH2+SPH2*SPS2
      T2N(3,2) = CPH2*SPS2*STH2-CPS2*SPH2
      T2N(3,3) = CPH2*CTH2
      TN2(1,1) = T2N(1,1)
      TN2(1,2) = T2N(2,1)
      TN2(1,3) = T2N(3,1)
      TN2(2,1) = T2N(1,2)
      TN2(2,2) = T2N(2,2)
      TN2(2,3) = T2N(3,2)
      TN2(3,1) = T2N(1,3)
      TN2(3,2) = T2N(2,3)
      TN2(3,3) = T2N(3,3)

C***  SEC. 200.  LOAD AERODYNAMICS.

C  load air velocity: v2sn = v1sn - TN1*S(r1sa1)*om11 -
TN2*S(ra2s2)*om22 +
C                             Tn2*dra2s2
C                   or v2sn = v1sn + A22*om11 + A23*om22 + dra2sn
      A22(1,1) = TN1(1,3)*R1SA1(2) - TN1(1,2)*R1SA1(3)
      A22(1,2) = TN1(1,1)*R1SA1(3) - R1SA1(1)*TN1(1,3)
      A22(1,3) = R1SA1(1)*TN1(1,2) - TN1(1,1)*R1SA1(2)
      A22(2,1) = R1SA1(2)*TN1(2,3) - TN1(2,2)*R1SA1(3)
      A22(2,2) = TN1(2,1)*R1SA1(3) - R1SA1(1)*TN1(2,3)
      A22(2,3) = R1SA1(1)*TN1(2,2) - R1SA1(2)*TN1(2,1)
      A22(3,1) = R1SA1(2)*TN1(3,3) - R1SA1(3)*TN1(3,2)
      A22(3,2) = R1SA1(3)*TN1(3,1) - R1SA1(1)*TN1(3,3)
      A22(3,3) = R1SA1(1)*TN1(3,2) - R1SA1(2)*TN1(3,1)
      A23(1,1) = TN2(1,3)*RA2S2(2) - TN2(1,2)*RA2S2(3)
```

```
      A23(1,2) = TN2(1,1)*RA2S2(3) - RA2S2(1)*TN2(1,3)
      A23(1,3) = RA2S2(1)*TN2(1,2) - TN2(1,1)*RA2S2(2)
      A23(2,1) = RA2S2(2)*TN2(2,3) - TN2(2,2)*RA2S2(3)
      A23(2,2) = TN2(2,1)*RA2S2(3) - RA2S2(1)*TN2(2,3)
      A23(2,3) = RA2S2(1)*TN2(2,2) - RA2S2(2)*TN2(2,1)
      A23(3,1) = RA2S2(2)*TN2(3,3) - RA2S2(3)*TN2(3,2)
      A23(3,2) = RA2S2(3)*TN2(3,1) - RA2S2(1)*TN2(3,3)
      A23(3,3) = RA2S2(1)*TN2(3,2) - RA2S2(2)*TN2(3,1)
      COR1N(1) = A22(1,3)*OM11(3)+A22(1,2)*OM11(2)+OM11(1)*A22(1,1)
      COR1N(2) = A22(2,3)*OM11(3)+OM11(2)*A22(2,2)+OM11(1)*A22(2,1)
      COR1N(3) = OM11(3)*A22(3,3)+OM11(2)*A22(3,2)+OM11(1)*A22(3,1)
      COR2N(1) = A23(1,3)*OM22(3)+A23(1,2)*OM22(2)+OM22(1)*A23(1,1)
      COR2N(2) = A23(2,3)*OM22(3)+OM22(2)*A23(2,2)+OM22(1)*A23(2,1)
      COR2N(3) = OM22(3)*A23(3,3)+OM22(2)*A23(3,2)+OM22(1)*A23(3,1)
      DRA2SN(1) = TN2(1,3)*DRA2S2(3)+TN2(1,2)*DRA2S2(2)+
     x            DRA2S2(1)*TN2(1,1)
      DRA2SN(2) = TN2(2,3)*DRA2S2(3)+DRA2S2(2)*TN2(2,2)+
     x            DRA2S2(1)*TN2(2,1)
      DRA2SN(3) = DRA2S2(3)*TN2(3,3)+DRA2S2(2)*TN2(3,2)+
     x            DRA2S2(1)*TN2(3,1)
      V1S2SN(1) = DRA2SN(1) + COR2N(1) + COR1N(1)
      V1S2SN(2) = DRA2SN(2) + COR2N(2) + COR1N(2)
      V1S2SN(3) = DRA2SN(3) + COR2N(3) + COR1N(3)
      V2SN(1)  = V1SN(1) + V1S2SN(1)
      V2SN(2)  = V1SN(2) + V1S2SN(2)
      V2SN(3)  = V1SN(3) + V1S2SN(3)
      VA2SN(1) = V2SN(1) - WN(1)
      VA2SN(2) = V2SN(2) - WN(2)
      VA2SN(3) = V2SN(3) - WN(3)
      VA2S2(1) = T2N(1,3)*VA2SN(3)+T2N(1,2)*VA2SN(2)+VA2SN(1)*T2N(1,1)
      VA2S2(2) = T2N(2,3)*VA2SN(3)+VA2SN(2)*T2N(2,2)+VA2SN(1)*T2N(2,1)
      VA2S2(3) = VA2SN(3)*T2N(3,3)+VA2SN(2)*T2N(3,2)+VA2SN(1)*T2N(3,1)


C******  LOAD AERODYNAMICS ****************
C Load aerodynamics:  option IAERSL = 0: no aero, 1: drag only, 2:
CONEX aero
      DO 110 I = 1,3
      FA22(I)  = 0.
 110  MA22(I)  = 0.
      IF (IWAKE.EQ.1) CALL WAKE
      IF (IAERSL.EQ.1) THEN
         VA2     = SQRT(VA2S2(1)**2 + VA2S2(2)**2 + VA2S2(3)**2)
         DOV     = DOQ*RHO*VA2/2.
         FA22(1) = -DOV*VA2S2(1)
         FA22(2) = -DOV*VA2S2(2)
         FA22(3) = -DOV*VA2S2(3)
      ELSE IF (IAERSL.EQ.2) THEN
         CALL CONEXAERO
      ELSE IF (IAERSL.EQ.3) THEN
         CALL CONEX4
         VA2     = SQRT(VA2S2(1)**2 + VA2S2(2)**2 + VA2S2(3)**2)
         RWIND   = 5*ABS(R2)
         DYNAM   = KDYN*SIGN(1,R2)*.5*RHO*VA2*(VA2 - RWIND)
         SWIRL   = MAX(0,KSWRL*(1-VA2/33.756))
         MA22(3) = MA22(3) + DYNAM + SWIRL
```

```
      ELSE IF (IAERSL.EQ.4) THEN
         CALL CONEX4A
         VA2     = SQRT(VA2S2(1)**2 + VA2S2(2)**2 + VA2S2(3)**2)
         RWIND   = 5*ABS(R2)
         DYNAM   = KDYN*SIGN(1,R2)*.5*RHO*VA2*(VA2 - RWIND)
         SWIRL   = MAX(0,KSWRL*(1-VA2/33.756))
         MA22(3) = MA22(3) + DYNAM + SWIRL

      END IF

C Call for Load Stabilization Program
      IF (STABSYST .EQ. 1) CALL LOADSTAB1

C***  SEC 300. fo = fg + fa - X - D dA u

      TV12(1) =  OM22(2)*RA2S2(3) - RA2S2(2)*OM22(3) + 2.*DRA2S2(1)
      TV12(2) = -OM22(1)*RA2S2(3) + RA2S2(1)*OM22(3) + 2.*DRA2S2(2)
      TV12(3) =  OM22(1)*RA2S2(2) - RA2S2(1)*OM22(2) + 2.*DRA2S2(3)
      TV22(1) =  OM22(2)*TV12(3) - TV12(2)*OM22(3)
      TV22(2) = -OM22(1)*TV12(3) + TV12(1)*OM22(3)
      TV22(3) =  OM22(1)*TV12(2) - TV12(1)*OM22(2)
      TV2N(1) = TN2(1,3)*TV22(3)+TN2(1,2)*TV22(2)+TV22(1)*TN2(1,1)
      TV2N(2) = TN2(2,3)*TV22(3)+TV22(2)*TN2(2,2)+TV22(1)*TN2(2,1)
      TV2N(3) = TV22(3)*TN2(3,3)+TV22(2)*TN2(3,2)+TV22(1)*TN2(3,1)
      CA11(1) = OM11(2)*(OM11(1)*R1SA1(2)-R1SA1(1)*OM11(2))-
     x          OM11(3)*(R1SA1(1)*OM11(3)-OM11(1)*R1SA1(3))
      CA11(2) = OM11(3)*(OM11(2)*R1SA1(3)-R1SA1(2)*OM11(3))-
     x          OM11(1)*(OM11(1)*R1SA1(2)-R1SA1(1)*OM11(2))
      CA11(3) = OM11(1)*(R1SA1(1)*OM11(3)-OM11(1)*R1SA1(3))-
     x          OM11(2)*(OM11(2)*R1SA1(3)-R1SA1(2)*OM11(3))
      DAU2N(1) = TN1(1,3)*CA11(3)+TN1(1,2)*CA11(2)+
     x           CA11(1)*TN1(1,1)+TV2N(1)
      DAU2N(2) = TN1(2,3)*CA11(3)+CA11(2)*TN1(2,2)+
     x           CA11(1)*TN1(2,1)+TV2N(2)
      DAU2N(3) = CA11(3)*TN1(3,3)+CA11(2)*TN1(3,2)+
     x           CA11(1)*TN1(3,1)+TV2N(3)
      JOM11(1) = OM11(1)*I1XX-OM11(3)*I1XZ
      JOM11(2) = OM11(2)*I1YY
      JOM11(3) = OM11(3)*I1ZZ-OM11(1)*I1XZ
      JOM22(1) = OM22(1)*I2XX-OM22(3)*I2XZ
      JOM22(2) = OM22(2)*I2YY
      JOM22(3) = OM22(3)*I2ZZ-OM22(1)*I2XZ
      X11(1)  = OM11(2)*JOM11(3) - JOM11(2)*OM11(3)
      X11(2)  = JOM11(1)*OM11(3) - OM11(1)*JOM11(3)
      X11(3)  = OM11(1)*JOM11(2) - JOM11(1)*OM11(2)
      X22(1)  = OM22(2)*JOM22(3) - JOM22(2)*OM22(3)
      X22(2)  = JOM22(1)*OM22(3) - OM22(1)*JOM22(3)
      X22(3)  = OM22(1)*JOM22(2) - JOM22(1)*OM22(2)
      FA1N(1) = TN1(1,3)*FA11(3)+TN1(1,2)*FA11(2)+FA11(1)*TN1(1,1)
      FA1N(2) = TN1(2,3)*FA11(3)+FA11(2)*TN1(2,2)+FA11(1)*TN1(2,1)
      FA1N(3) = FA11(3)*TN1(3,3)+FA11(2)*TN1(3,2)+FA11(1)*TN1(3,1)

C  add in the weight of the rotor system to the total aerodynamic
C  force on the system because GENHEL treats the helo/rotor as two a
C  two body system, but the ghslmc logic requires entire helo as one
body
```

```
C  and the load as the second.
      FA1N(3) = FA1N(3) - WBLADE*NBS
      FA11(1) = T1N(1,3)*FA1N(3)+T1N(1,2)*FA1N(2)+FA1N(1)*T1N(1,1)
      FA11(2) = T1N(2,3)*FA1N(3)+FA1N(2)*T1N(2,2)+FA1N(1)*T1N(2,1)
      FA11(3) = FA1N(3)*T1N(3,3)+FA1N(2)*T1N(3,2)+FA1N(1)*T1N(3,1)

      FA2N(1) = TN2(1,3)*FA22(3)+TN2(1,2)*FA22(2)+FA22(1)*TN2(1,1)
      FA2N(2) = TN2(2,3)*FA22(3)+FA22(2)*TN2(2,2)+FA22(1)*TN2(2,1)
      FA2N(3) = FA22(3)*TN2(3,3)+FA22(2)*TN2(3,2)+FA22(1)*TN2(3,1)
      FO1N(1) = FA1N(1)
      FO1N(2) = FA1N(2)
      FO1N(3) = W1 + FA1N(3)
      FO2N(1) = FA2N(1) - M2*DAU2N(1)
      FO2N(2) = FA2N(2) - M2*DAU2N(2)
      FO2N(3) = W2 + FA2N(3) - M2*DAU2N(3)
      MO11(1) = MA11(1) - X11(1)
      MO11(2) = MA11(2) - X11(2)
      MO11(3) = MA11(3) - X11(3)
      MO22(1) = MA22(1) - X22(1)
      MO22(2) = MA22(2) - X22(2)
      MO22(3) = MA22(3) - X22(3)

C***  SEC 400:  SUSPENSION FORCES ON HC AND LOAD     fc = H fc1n

      IF (STRETCH.EQ.0) THEN

C  inelastic cable; fc1n = -[HT*DI*H]^-1*HT*DI*fo
         A22J1I(1,1) = A22(1,3)*II1XZ+A22(1,1)*II1XX
         A22J1I(1,2) = A22(1,2)*II1YY
         A22J1I(1,3) = A22(1,3)*II1ZZ+A22(1,1)*II1XZ
         A22J1I(2,1) = A22(2,3)*II1XZ+A22(2,1)*II1XX
         A22J1I(2,2) = A22(2,2)*II1YY
         A22J1I(2,3) = A22(2,3)*II1ZZ+A22(2,1)*II1XZ
         A22J1I(3,1) = A22(3,3)*II1XZ+A22(3,1)*II1XX
         A22J1I(3,2) = A22(3,2)*II1YY
         A22J1I(3,3) = A22(3,3)*II1ZZ+A22(3,1)*II1XZ
         A23J2I(1,1) = A23(1,3)*II2XZ+A23(1,1)*II2XX
         A23J2I(1,2) = A23(1,2)*II2YY
         A23J2I(1,3) = A23(1,3)*II2ZZ+A23(1,1)*II2XZ
         A23J2I(2,1) = A23(2,3)*II2XZ+A23(2,1)*II2XX
         A23J2I(2,2) = A23(2,2)*II2YY
         A23J2I(2,3) = A23(2,3)*II2ZZ+A23(2,1)*II2XZ
         A23J2I(3,1) = A23(3,3)*II2XZ+A23(3,1)*II2XX
         A23J2I(3,2) = A23(3,2)*II2YY
         A23J2I(3,3) = A23(3,3)*II2ZZ+A23(3,1)*II2XZ
         TV3N(1) = A22J1I(1,3)*MO11(3)+A22J1I(1,2)*MO11(2)+
     x             MO11(1)*A22J1I(1,1)
         TV3N(2) = A22J1I(2,3)*MO11(3)+MO11(2)*A22J1I(2,2)+
     x             MO11(1)*A22J1I(2,1)
         TV3N(3) = MO11(3)*A22J1I(3,3)+MO11(2)*A22J1I(3,2)+
     x             MO11(1)*A22J1I(3,1)
         TV4N(1) = A23J2I(1,3)*MO22(3)+A23J2I(1,2)*MO22(2)+
     x             MO22(1)*A23J2I(1,1)
         TV4N(2) = A23J2I(2,3)*MO22(3)+MO22(2)*A23J2I(2,2)+
     x             MO22(1)*A23J2I(2,1)
         TV4N(3) = MO22(3)*A23J2I(3,3)+MO22(2)*A23J2I(3,2)+
     x             MO22(1)*A23J2I(3,1)
```

```
      HTDIFO(1) = -FO2N(1)/M2+FO1N(1)/M1+TV4N(1)+TV3N(1)
      HTDIFO(2) = -FO2N(2)/M2+FO1N(2)/M1+TV4N(2)+TV3N(2)
      HTDIFO(3) = -FO2N(3)/M2+FO1N(3)/M1+TV4N(3)+TV3N(3)
      RJR1(1,1) = A22(1,3)*A22J1I(1,3)+A22(1,2)*A22J1I(1,2)+
    x             A22(1,1)*A22J1I(1,1)
      RJR1(1,2) = A22J1I(1,3)*A22(2,3)+A22J1I(1,2)*A22(2,2)+
    x             A22J1I(1,1)*A22(2,1)
      RJR1(1,3) = A22J1I(1,3)*A22(3,3)+A22J1I(1,2)*A22(3,2)+
    x             A22J1I(1,1)*A22(3,1)
      RJR1(2,2) = A22(2,3)*A22J1I(2,3)+A22(2,2)*A22J1I(2,2)+
    x             A22(2,1)*A22J1I(2,1)
      RJR1(2,3) = A22J1I(2,3)*A22(3,3)+A22J1I(2,2)*A22(3,2)+
    x             A22J1I(2,1)*A22(3,1)
      RJR1(3,3) = A22(3,3)*A22J1I(3,3)+A22(3,2)*A22J1I(3,2)+
    x             A22(3,1)*A22J1I(3,1)
      RJR2(1,1) = A23(1,3)*A23J2I(1,3)+A23(1,2)*A23J2I(1,2)+
    x             A23(1,1)*A23J2I(1,1)
      RJR2(1,2) = A23J2I(1,3)*A23(2,3)+A23J2I(1,2)*A23(2,2)+
    x             A23J2I(1,1)*A23(2,1)
      RJR2(1,3) = A23J2I(1,3)*A23(3,3)+A23J2I(1,2)*A23(3,2)+
    x             A23J2I(1,1)*A23(3,1)
      RJR2(2,2) = A23(2,3)*A23J2I(2,3)+A23(2,2)*A23J2I(2,2)+
    x             A23(2,1)*A23J2I(2,1)
      RJR2(2,3) = A23J2I(2,3)*A23(3,3)+A23J2I(2,2)*A23(3,2)+
    x             A23J2I(2,1)*A23(3,1)
      RJR2(3,3) = A23(3,3)*A23J2I(3,3)+A23(3,2)*A23J2I(3,2)+
    x             A23(3,1)*A23J2I(3,1)

C  upper triangle elements of symmetric HTDIH
      S11 = MU12+RJR2(1,1)+RJR1(1,1)
      S12 = RJR2(1,2)+RJR1(1,2)
      S13 = RJR2(1,3)+RJR1(1,3)
      S22 = MU12+RJR2(2,2)+RJR1(2,2)
      S23 = RJR2(2,3)+RJR1(2,3)
      S33 = MU12+RJR2(3,3)+RJR1(3,3)

C  upper triangle elements of symmetric HTDIH^-1
      CF11 = S22*S33 - S23*S23
      CF12 = S12*S33 - S13*S23
      CF13 = S12*S23 - S13*S22
      CF22 = S11*S33 - S13*S13
      CF23 = S23*S11 - S12*S13
      CF33 = S11*S22 - S12*S12
      DET  = S11*CF11 - S12*CF12 + S13*CF13
      SI11 =  CF11/DET
      SI12 = -CF12/DET
      SI13 =  CF13/DET
      SI22 =  CF22/DET
      SI23 = -CF23/DET
      SI33 =  CF33/DET

C  cable force on helicopter, inertial axes components
      FC1N(1) = -HTDIFO(3)*SI13-HTDIFO(2)*SI12-HTDIFO(1)*SI11
      FC1N(2) = -HTDIFO(3)*SI23-HTDIFO(2)*SI22-HTDIFO(1)*SI12
      FC1N(3) = -HTDIFO(3)*SI33-HTDIFO(2)*SI23-HTDIFO(1)*SI13

   ELSE
```

```
C  elastic cables: fc12 = sum(tauj*kcj2, j = 1,...,nc)

         FC12(1) = 0.
         FC12(2) = 0.
         FC12(3) = 0.

         DO 300 J = 1,NC
         RAJ2(1,J) = RA2S2(1) + R2SJ2(1,J)
         RAJ2(2,J) = RA2S2(2) + R2SJ2(2,J)
         RAJ2(3,J) = RA2S2(3) + R2SJ2(3,J)
         LCJ(J)    = SQRT(RAJ2(3,J)**2+RAJ2(2,J)**2+RAJ2(1,J)**2)
         KCJ2(1,J) = RAJ2(1,J)/LCJ(J)
         KCJ2(2,J) = RAJ2(2,J)/LCJ(J)
         KCJ2(3,J) = RAJ2(3,J)/LCJ(J)
         DLCJ(J)   = DRA2S2(3)*KCJ2(3,J)+DRA2S2(2)*KCJ2(2,J)
     X               +DRA2S2(1)*KCJ2(1,J)
         TAUJ(J)   = AMAX1(0.,KS*(LCJ(J) - LCJO(J)) + CS*DLCJ(J))
         FC12(1)   = FC12(1) + KCJ2(1,J)*TAUJ(J)
         FC12(2)   = FC12(2) + KCJ2(2,J)*TAUJ(J)
         FC12(3)   = FC12(3) + KCJ2(3,J)*TAUJ(J)
 300     CONTINUE

         FC1N(1) = TN2(1,3)*FC12(3)+TN2(1,2)*FC12(2)+FC12(1)*TN2(1,1)
         FC1N(2) = TN2(2,3)*FC12(3)+FC12(2)*TN2(2,2)+FC12(1)*TN2(2,1)
         FC1N(3) = FC12(3)*TN2(3,3)+FC12(2)*TN2(3,2)+FC12(1)*TN2(3,1)

      ENDIF

C  in sim IC use susp force result from load-susp IC logic
      IF (IMODE.LT.0) THEN
         FC1N(1) = -TN2(1,3)*FC22(3)-TN2(1,2)*FC22(2)-FC22(1)*TN2(1,1)
         FC1N(2) = -TN2(2,3)*FC22(3)-FC22(2)*TN2(2,2)-FC22(1)*TN2(2,1)
         FC1N(3) = -FC22(3)*TN2(3,3)-FC22(2)*TN2(3,2)-FC22(1)*TN2(3,1)
      END IF

C  suspension forces and cg moments on the HC and load

      FC2N(1) = -FC1N(1)
      FC2N(2) = -FC1N(2)
      FC2N(3) = -FC1N(3)
      MC11(1) = FC1N(3)*A22(3,1)+FC1N(2)*A22(2,1)+FC1N(1)*A22(1,1)
      MC11(2) = FC1N(3)*A22(3,2)+FC1N(2)*A22(2,2)+FC1N(1)*A22(1,2)
      MC11(3) = FC1N(3)*A22(3,3)+FC1N(2)*A22(2,3)+FC1N(1)*A22(1,3)
      MC22(1) = FC1N(3)*A23(3,1)+FC1N(2)*A23(2,1)+FC1N(1)*A23(1,1)
      MC22(2) = FC1N(3)*A23(3,2)+FC1N(2)*A23(2,2)+FC1N(1)*A23(1,2)
      MC22(3) = FC1N(3)*A23(3,3)+FC1N(2)*A23(2,3)+FC1N(1)*A23(1,3)


C Addition of yaw resistence torque at the hook
      DELPS   = PS2 - PS1
      SLWND   = .98*(1 - 6.283/(5*ABS(DELPS) + .017))
      CALL SLINGTORQ
      YMHOOK  = -KR2*R2 - KPS
      MC22(3) = MC22(3) + YMHOOK
      K21(1)  = T1N(1,3)*K2N(3)+T1N(1,2)*K2N(2)+K2N(1)*T1N(1,1)
      K21(2)  = T1N(2,3)*K2N(3)+K2N(2)*T1N(2,2)+K2N(1)*T1N(2,1)
```

```
      K21(3)  = K2N(3)*T1N(3,3)+K2N(2)*T1N(3,2)+K2N(1)*T1N(3,1)
      MC11(1) = MC11(1) - YMHOOK*K21(1)
      MC11(2) = MC11(2) - YMHOOK*K21(2)
      MC11(3) = MC11(3) - YMHOOK*K21(3)



C***  SEC 500: ACCELERATIONS.  du = A^-1 D^-1 (fo + fc) = A^-1 sf

      SF1N(1)  = (FO1N(1) + FC1N(1))/M1
      SF1N(2)  = (FO1N(2) + FC1N(2))/M1
      SF1N(3)  = (FO1N(3) + FC1N(3))/M1
      SF2N(1)  = (FO2N(1) + FC2N(1))/M2
      SF2N(2)  = (FO2N(2) + FC2N(2))/M2
      SF2N(3)  = (FO2N(3) + FC2N(3))/M2
      SUMM11(1) = MO11(1) + MC11(1)
      SUMM11(2) = MO11(2) + MC11(2)
      SUMM11(3) = MO11(3) + MC11(3)
      SUMM22(1) = MO22(1) + MC22(1)
      SUMM22(2) = MO22(2) + MC22(2)
      SUMM22(3) = MO22(3) + MC22(3)
      SM11(1)  = SUMM11(3)*II1XZ + SUMM11(1)*II1XX
      SM11(2)  = SUMM11(2)*II1YY
      SM11(3)  = SUMM11(3)*II1ZZ + SUMM11(1)*II1XZ
      SM22(1)  = SUMM22(3)*II2XZ + SUMM22(1)*II2XX
      SM22(2)  = SUMM22(2)*II2YY
      SM22(3)  = SUMM22(3)*II2ZZ + SUMM22(1)*II2XZ
      DV1SN(1) = SF1N(1)
      DV1SN(2) = SF1N(2)
      DV1SN(3) = SF1N(3)
      DOM11(1) = SM11(1)
      DOM11(2) = SM11(2)
      DOM11(3) = SM11(3)
      DOM22(1) = SM22(1)
      DOM22(2) = SM22(2)
      DOM22(3) = SM22(3)
      TV5N(1)  = A22(1,3)*SM11(3)+A22(1,2)*SM11(2)+SM11(1)*A22(1,1)
      TV5N(2)  = A22(2,3)*SM11(3)+SM11(2)*A22(2,2)+SM11(1)*A22(2,1)
      TV5N(3)  = SM11(3)*A22(3,3)+SM11(2)*A22(3,2)+SM11(1)*A22(3,1)
      TV6N(1)  = A23(1,3)*SM22(3)+A23(1,2)*SM22(2)+SM22(1)*A23(1,1)
      TV6N(2)  = A23(2,3)*SM22(3)+SM22(2)*A23(2,2)+SM22(1)*A23(2,1)
      TV6N(3)  = SM22(3)*A23(3,3)+SM22(2)*A23(3,2)+SM22(1)*A23(3,1)
      DDRA2SN(1) = SF2N(1) - TV6N(1) - TV5N(1) - SF1N(1)
      DDRA2SN(2) = SF2N(2) - TV6N(2) - TV5N(2) - SF1N(2)
      DDRA2SN(3) = SF2N(3) - TV6N(3) - TV5N(3) - SF1N(3)
      DDRA2S2(1) = T2N(1,3)*DDRA2SN(3)+T2N(1,2)*DDRA2SN(2)+
     x             DDRA2SN(1)*T2N(1,1)
      DDRA2S2(2) = T2N(2,3)*DDRA2SN(3)+DDRA2SN(2)*T2N(2,2)+
     x             DDRA2SN(1)*T2N(2,1)
      DDRA2S2(3) = DDRA2SN(3)*T2N(3,3)+DDRA2SN(2)*T2N(3,2)+
     x             DDRA2SN(1)*T2N(3,1)

C***  SEC 600.  system cg errors,  initialize integrators

      R1S2SN(1) = TN2(1,3)*RA2S2(3)+TN1(1,3)*R1SA1(3)+
     x            TN2(1,2)*RA2S2(2)+TN1(1,2)*R1SA1(2)+
     x            RA2S2(1)*TN2(1,1)+R1SA1(1)*TN1(1,1)
```

```
      R1S2SN(2) = TN2(2,3)*RA2S2(3)+TN1(2,3)*R1SA1(3)+
     x            RA2S2(2)*TN2(2,2)+R1SA1(2)*TN1(2,2)+
     x            RA2S2(1)*TN2(2,1)+R1SA1(1)*TN1(2,1)
      R1S2SN(3) = RA2S2(3)*TN2(3,3)+R1SA1(3)*TN1(3,3)+
     x            RA2S2(2)*TN2(3,2)+R1SA1(2)*TN1(3,2)+
     x            RA2S2(1)*TN2(3,1)+R1SA1(1)*TN1(3,1)
      RSN(1) = R1S2SN(1)*M2OM + R1SN(1)
      RSN(2) = R1S2SN(2)*M2OM + R1SN(2)
      RSN(3) = R1S2SN(3)*M2OM + R1SN(3)
      VSN(1) = V1S2SN(1)*M2OM + V1SN(1)
      VSN(2) = V1S2SN(2)*M2OM + V1SN(2)
      VSN(3) = V1S2SN(3)*M2OM + V1SN(3)
      TV7N(1) = A22(1,3)*DOM11(3)+A22(1,2)*DOM11(2)+DOM11(1)*A22(1,1)
      TV7N(2) = A22(2,3)*DOM11(3)+DOM11(2)*A22(2,2)+DOM11(1)*A22(2,1)
      TV7N(3) = DOM11(3)*A22(3,3)+DOM11(2)*A22(3,2)+DOM11(1)*A22(3,1)
      TV8N(1) = A23(1,3)*DOM22(3)+A23(1,2)*DOM22(2)+DOM22(1)*A23(1,1)
      TV8N(2) = A23(2,3)*DOM22(3)+DOM22(2)*A23(2,2)+DOM22(1)*A23(2,1)
      TV8N(3) = DOM22(3)*A23(3,3)+DOM22(2)*A23(3,2)+DOM22(1)*A23(3,1)
      ADU2N(1) = TV8N(1) + TV7N(1) + DDRA2SN(1)
      ADU2N(2) = TV8N(2) + TV7N(2) + DDRA2SN(2)
      ADU2N(3) = TV8N(3) + TV7N(3) + DDRA2SN(3)
      DV1S2SN(1) = DAU2N(1) + ADU2N(1)
      DV1S2SN(2) = DAU2N(2) + ADU2N(2)
      DV1S2SN(3) = DAU2N(3) + ADU2N(3)
      DVSN(1) = DV1S2SN(1)*M2OM + DV1SN(1)
      DVSN(2) = DV1S2SN(2)*M2OM + DV1SN(2)
      DVSN(3) = DV1S2SN(3)*M2OM + DV1SN(3)
      SSFN(1) = (FA2N(1) + FA1N(1))/M1PM2
      SSFN(2) = (FA2N(2) + FA1N(2))/M1PM2
      SSFN(3) = (FA2N(3) + FA1N(3))/M1PM2 + G

C  save initial c.g. states and initialize integrations
      IF (T.LE.DTO2) THEN
         DO 600 I = 1,3
         RSON(I)    = RSN(I)
         VSON(I)    = VSN(I)
         ISSFN(I)   = 0.
         IISSFN(I)  = 0.
         OSSFN(I)   = SSFN(I)
         OISSFN(I)  = 0.
         DQ(I)      = V1SN(I)
 600     DQ(9+I)    = DRA2S2(I)

         DPS1       = (OM11(2)*SPH1 + OM11(3)*CPH1)/CTH1
         DPH1       = OM11(1) + DPS1*STH1
         DTH1       = OM11(2)*CPH1 - OM11(3)*SPH1
         DPS2       = (OM22(2)*SPH2 + OM22(3)*CPH2)/CTH2
         DPH2       = OM22(1) + DPS2*STH2
         DTH2       = OM22(2)*CPH2 - OM22(3)*SPH2

         DO 601 I = 1,12
         ODU(I)     = DU(I)
 601     ODQ(I)     = DQ(I)

      END IF

C  cg errors and integrate SSFN for cg errors
```

```
      DELDVSN(1) = DVSN(1) - SSFN(1)
      DELDVSN(2) = DVSN(2) - SSFN(2)
      DELDVSN(3) = DVSN(3) - SSFN(3)
      DELVSN(1)  = VSN(1) - ISSFN(1) - VSON(1)
      DELVSN(2)  = VSN(2) - ISSFN(2) - VSON(2)
      DELVSN(3)  = VSN(3) - ISSFN(3) - VSON(3)
      DELRSN(1)  = RSN(1) - VSON(1)*T - RSON(1) - IISSFN(1)
      DELRSN(2)  = RSN(2) - VSON(2)*T - RSON(2) - IISSFN(2)
      DELRSN(3)  = RSN(3) - VSON(3)*T - RSON(3) - IISSFN(3)

      DO 603 I = 1,3
      ISSFN(I)  = (3*SSFN(I) - OSSFN(I))*DTO2 + ISSFN(I)
      IISSFN(I) = (OISSFN(I) + ISSFN(I))*DTO2 + IISSFN(I)
      OSSFN(I)  = SSFN(I)
 603  OISSFN(I) = ISSFN(I)

C***  SEC 700.  transfer data to calling program and data storage


      FC11(1) = T1N(1,3)*FC1N(3)+T1N(1,2)*FC1N(2)+FC1N(1)*T1N(1,1)
      FC11(2) = T1N(2,3)*FC1N(3)+FC1N(2)*T1N(2,2)+FC1N(1)*T1N(2,1)
      FC11(3) = FC1N(3)*T1N(3,3)+FC1N(2)*T1N(3,2)+FC1N(1)*T1N(3,1)
      SLINGTENSION = SQRT(FC1N(1)**2 + FC1N(2)**2 + FC1N(3)**2)

C*** Remainder of this section generates auxilliary variables.

      DV1S1(1) = T1N(1,3)*DV1SN(3)+T1N(1,2)*DV1SN(2)+DV1SN(1)*T1N(1,1)
      DV1S1(2) = T1N(2,3)*DV1SN(3)+DV1SN(2)*T1N(2,2)+DV1SN(1)*T1N(2,1)
      DV1S1(3) = DV1SN(3)*T1N(3,3)+DV1SN(2)*T1N(3,2)+DV1SN(1)*T1N(3,1)

      FC22(1) = T2N(1,3)*FC2N(3)+T2N(1,2)*FC2N(2)+FC2N(1)*T2N(1,1)
      FC22(2) = T2N(2,3)*FC2N(3)+FC2N(2)*T2N(2,2)+FC2N(1)*T2N(2,1)
      FC22(3) = FC2N(3)*T2N(3,3)+FC2N(2)*T2N(3,2)+FC2N(1)*T2N(3,1)

C  hook-to-load-cg vector
         RA2SN(1)   = TN2(1,3)*RA2S2(3)+TN2(1,2)*RA2S2(2)+
     x                RA2S2(1)*TN2(1,1)
         RA2SN(2)   = TN2(2,3)*RA2S2(3)+RA2S2(2)*TN2(2,2)+
     x                RA2S2(1)*TN2(2,1)
         RA2SN(3)   = RA2S2(3)*TN2(3,3)+RA2S2(2)*TN2(3,2)+
     x                RA2S2(1)*TN2(3,1)
         RA2S1(1) = T1N(1,3)*RA2SN(3)+T1N(1,2)*RA2SN(2)+
     x              RA2SN(1)*T1N(1,1)
         RA2S1(2) = T1N(2,3)*RA2SN(3)+RA2SN(2)*T1N(2,2)+
     x              RA2SN(1)*T1N(2,1)
         RA2S1(3) = RA2SN(3)*T1N(3,3)+RA2SN(2)*T1N(3,2)+
     x              RA2SN(1)*T1N(3,1)

C  load inertial position (for possible 2D, 3D trajectory plots)
         R2SN(1) = R1SN(1) + R1S2SN(1)
         R2SN(2) = R1SN(2) + R1S2SN(2)
         R2SN(3) = R1SN(3) + R1S2SN(3)

C  load acceleration, body axes components
         DV2SN(1) = DV1SN(1) + DV1S2SN(1)
         DV2SN(2) = DV1SN(2) + DV1S2SN(2)
         DV2SN(3) = DV1SN(3) + DV1S2SN(3)
```

```
         DV2S2(1) = T2N(1,3)*DV2SN(3)+T2N(1,2)*DV2SN(2)+
     x              DV2SN(1)*T2N(1,1)
         DV2S2(2) = T2N(2,3)*DV2SN(3)+DV2SN(2)*T2N(2,2)+
     x              DV2SN(1)*T2N(2,1)
         DV2S2(3) = DV2SN(3)*T2N(3,3)+DV2SN(2)*T2N(3,2)+
     x              DV2SN(1)*T2N(3,1)


C load accelerometer readings

      DV2SS2(1) =-(Q2*Q2+R2*R2)*R2SS2(1) + (P2*Q2-DR2)*R2SS2(2)
     X      + (P2*R2+DQ2)*R2SS2(3)
      DV2SS2(2) = (P2*Q2+DR2)*R2SS2(1) - (P2*P2+R2*R2)*R2SS2(2)
     X      + (Q2*R2-DP2)*R2SS2(3)
      DV2SS2(3) = (P2*R2-DQ2)*R2SS2(1) + (Q2*R2+DP2)*R2SS2(2)
     X      -(P2*P2+Q2*Q2)*R2SS2(3)
      AMGS2(1) = DV2S2(1) - G*KN2(1) + DV2SS2(1)
      AMGS2(2) = DV2S2(2) - G*KN2(2) + DV2SS2(2)
      AMGS2(3) = DV2S2(3) - G*KN2(3) + DV2SS2(3)

C  load p, q in load-HC heading axes per MT
         CPS2M1 = COS(PS2 - PS1)
         SPS2M1 = SIN(PS2 - PS1)
         P2P    = OM22(1)*CPS2M1 - OM22(2)*SPS2M1
         Q2P    = OM22(1)*SPS2M1 + OM22(2)*CPS2M1

C pht 3 mar 99 transformed load motion:
C load ang vel in inertial coords, and in level heading (?) coords

      CPS2M1    = COS(PS2 - PS1)
      SPS2M1    = SIN(PS2 - PS1)
      SPH2M1    = SIN(PH2)
      CPH2M1    = COS(PH2)
      STH2M1    = SIN(TH2)
      CTH2M1    = COS(TH2)
      T1T2(1,1) = CPS2M1*CTH2M1
      T1T2(2,1) = CTH2M1*SPS2M1
      T1T2(3,1) = -STH2M1
      T1T2(1,2) = CPS2M1*SPH2M1*STH2M1 - CPH2M1*SPS2M1
      T1T2(2,2) = SPH2M1*SPS2M1*STH2M1 + CPH2M1*CPS2M1
      T1T2(3,2) = CTH2M1*SPH2M1
      T1T2(1,3) = CPH2M1*CPS2M1*STH2M1 + SPH2M1*SPS2M1
      T1T2(2,3) = CPH2M1*SPS2M1*STH2M1 - CPS2M1*SPH2M1
      T1T2(3,3) = CPH2M1*CTH2M1

      P2N = TN2(1,3)*OM22(3) + TN2(1,2)*OM22(2) + OM22(1)*TN2(1,1)
      Q2N = TN2(2,3)*OM22(3) + OM22(2)*TN2(2,2) + OM22(1)*TN2(2,1)
      R2N = OM22(3)*TN2(3,3) + OM22(2)*TN2(3,2) + OM22(1)*TN2(3,1)

      P21 = T1T2(1,3)*OM22(3) + OM22(2)*T1T2(1,2) + OM22(1)*T1T2(1,1)
      Q21 = T1T2(2,3)*OM22(3) + OM22(2)*T1T2(2,2) + OM22(1)*T1T2(2,1)
      R21 = T1T2(3,3)*OM22(3) + OM22(2)*T1T2(3,2) + OM22(1)*T1T2(3,1)
C end pht

         NS = NS + 1
```

```
C***  SEC 800. INTEGRATION: compute u, q at tn + dt, store past values

C  update u.  null stretching rate exactly if inelastic suspension

      DO 800 I = 1,12
 800  U(I) = U(I) + (3*DU(I) - ODU(I))*DTO2
      IF (STRETCH.EQ.0) THEN
         DRA2S2(1) = 0.
         DRA2S2(2) = 0.
         DRA2S2(3) = 0.
      END IF

C  compute dq(q,u) using updated u
      DO 801 I = 1,3
      DQ(I)   = V1SN(I)
 801  DQ(I+9) = DRA2S2(I)

      DPS1   = (OM11(2)*SPH1 + OM11(3)*CPH1)/CTH1
      DPH1   = OM11(1) + DPS1*STH1
      DTH1   = OM11(2)*CPH1 - OM11(3)*SPH1
      DPS2   = (OM22(2)*SPH2 + OM22(3)*CPH2)/CTH2
      DPH2   = OM22(1) + DPS2*STH2
      DTH2   = OM22(2)*CPH2 - OM22(3)*SPH2

C  update q and store past values
      DO 802 I = 1,12
      Q(I)   = Q(I) + (ODQ(I) + DQ(I))*DTO2
      ODU(I) = DU(I)
 802  ODQ(I) = DQ(I)

      END
```

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX K GHSL_INIT.F

Initialization module for GenHel / Slung Load simulation. Load stability parameters were added. . Program by Tyson, P. modified by Ehlers, G.

```
C file genhel/batch/sl/ghsl_init.f  -------------------------  31 AUG
98 Peter Tyson
C    Load Stabil Parameters added Jul 01 by George Ehlers

C     Initialization for GenHel/Slung Load Simulation.
C  Reads input data from 'ghsl.dat' which resides in the GenHel/batch
C  directory.  Load Configuration and Slung Load run parameters are
C  appended to the FLTNAME//01.INFO outfile (ASCII format) and to the
C  screen dump during nrt program operation (only for SLSC or SLMC
cases).
C  Subroutine is called by BHAWK_NRT_EXEC immediately following call
C  for BHAWK_NRT_INIT.

      SUBROUTINE GHSL_INIT

      INCLUDE 'slvars.cmn'

      REAL FSHK, BLHK, WLHK, FSHB, BLHB, WLHB, FSCGTNK, FSWI, FWMX,
     X     ESBLCG, ESWLCG, AIRSPEED, KSWRL, KR2, KPS2, KDYN
      CHARACTER CABLES

      EQUIVALENCE (A(238), AIRSPEED)

      DATA
     X   ICUNIT, IOUT     /1, 6/
     X   FSHK, WLHK, BLHK /352.6, 195.5, 0./,
     X   FSHB, WLHB, BLHB /341.2, 315.0, 0./,
     X   FSCGTNK, FSWI    /420.8, 6.8/,
     X   ESBLCG, ESWLCG   /0.0,  247.2/

      NAMELIST /HCDATA/ TOW, XMOMTO, FWT, FWMX, AIRSPEED

      NAMELIST /SLRUN/ CHFILE, AXIS, DATAFILE, STRETCH,
     X   IAERSL, ILOAD, IPILOT, IWAKE, ISWIRL, IDATA, STABSYST,
     X   DYNAMIC, PWL_TIME, PWLCNTRL


c123456789c123456789c123456789c123456789c123456789c123456789c123456789c
1

      NAMELIST /SLDATA/ LOADNAME, W2, I2XX, I2YY, I2ZZ, I2XZ,  KS,
     X   CS, LCO, RA2PO2, R2P2SO2, R2P2S2, NC, R2PJ2, R2S12, DOQ,
     X   DELPS2O, R2OD, R2SS2, KDYN, KSWRL, KR2, KPS2

C Load stabilization Parameters

      NAMELIST /STABDAT/ ACTIVE, TEMP, GAINH, CH, SH, CLAH, CLDH,
     X  CDOH, CDPH, STALH1, STALH2, RH, GAINV, CV, SV, CLAV, CLDV,
```

```
     X  CDOV, CDPV, STALV1, STALV2,  RV

C reset RLOAD common block variables
      DO 1 I = 1,500
 1    RLOAD(I) = 0
      NREC     = 0

      OPEN(ICUNIT,FILE='ghsl.dat',STATUS='OLD',READONLY,ERR=14)
      READ(ICUNIT,HCDATA)
      REWIND ICUNIT
      READ(ICUNIT,SLRUN)
      REWIND ICUNIT
      READ(ICUNIT,SLDATA)
      CLOSE(ICUNIT)
      OPEN(ICUNIT,FILE='loadstab.dat',STATUS='OLD',READONLY,ERR=15)
      READ(ICUNIT,STABDAT)
      CLOSE(ICUNIT)

      XMOMI  = XMOMTO - FWMX*FSCGTNK
      ZMOMI  = TOW*ESWLCG - (204.75 + .09116*FWMX/FSWI)*FWMX
      W1     = TOW - FWMX + FWT
      FSCG   = (XMOMI + FWT*FSCGTNK)/W1
      WLCG   = (ZMOMI + (204.75 + .09116*FWT/FSWI)*FWT)/W1
      BLCG   = ESBLCG

C--- position vector, HC c.g. to hook, HC bocy axes ---C

      R1SA1(1)  = (FSCG - FSHK)/12.
      R1SA1(2)  = (-BLCG + BLHK)/12.
      R1SA1(3)  = (WLCG - WLHK)/12.

      IF (CHFILE.NE.'sweep.dat') AXIS = 1

      IF (ILOAD.EQ.0) WRITE(IOUT,100)
      IF (ILOAD.EQ.1) THEN
         WRITE(IOUT,101)
         CABLES = ' '
      END IF
      IF (ILOAD.EQ.2) THEN
         WRITE(IOUT,102)
         CABLES = 'S'
      END IF
      IF (ILOAD.NE.0) THEN
         WRITE(IOUT,103) LOADNAME, W2, I2XX, I2YY, I2ZZ, I2XZ, RA2PO2,
     X      R2P2SO2, R2P2S2, (R2PJ2(1,J),R2PJ2(2,J),R2PJ2(3,J),J=1,NC)
         WRITE(IOUT,104) TOW, W1, FWT
         WRITE(IOUT,105) KS, CS, DT
         IF (STRETCH.EQ.0) WRITE(IOUT,106) CABLES
         IF (STRETCH.EQ.1) WRITE(IOUT,107) CABLES
         IF ((IAERSL.EQ.1).AND.(DOQ.EQ.0.)) WRITE (IOUT,108)
         IF (IAERSL.EQ.1) WRITE(IOUT,109)
         IF (IAERSL.EQ.2) WRITE(IOUT,110)
         IF (IWAKE.EQ.0)  WRITE(IOUT,111)
         IF (IWAKE.EQ.1)  THEN
            IF (ISWIRL.EQ.0) WRITE(IOUT,112)
            IF (ISWIRL.EQ.1) WRITE(IOUT,113)
         END IF
```

```
      END IF
      IF (IPILOT.EQ.0) WRITE(IOUT,114)
      IF (IPILOT.EQ.1) WRITE(IOUT,115)
      IF (CHFILE.EQ.'sweep.dat') THEN
         IF (AXIS.EQ.1) WRITE(IOUT,116)
         IF (AXIS.EQ.2) WRITE(IOUT,117)
         IF (AXIS.EQ.3) WRITE(IOUT,118)
         IF (AXIS.EQ.4) WRITE(IOUT,119)
      END IF
      RETURN

100   FORMAT(5X'NO LOAD SIMULATION')
101   FORMAT(5X'SINGLE LOAD, SINGLE CABLE SIMULATION')
102   FORMAT(5X'SINGLE LOAD, MULTI-CABLE SIMULATION')
103   FORMAT(//
     X'                     L O A D   C O N F I G U R A T I O N'//
     X5X'LOAD DESCRIPTION :  'A//
     X5X'LOAD WEIGHT      :  'F8.2//
     X5X'LOAD INERTIA XX  :  'F8.2,
     X'  LOAD INERTIA YY  :  'F8.2,'   LB-FT-S**2',/
     X5X'LOAD INERTIA ZZ  :  'F8.2,
     X'  LOAD INERTIA XZ  :  'F8.2,'   LB-FT-S**2',//
     X5X'RA2PO2   :  '3F8.2/
     X5X'R2P2SO2  :  '3F8.2/
     X5X'R2P2S2   :  '3F8.2//
     X5X'R2PJ2    :  '/8(10X,3F8.2/))
104   FORMAT(5X'T/O WT, CURRENT WT, FUEL WT:  '3F10.2/)
105   FORMAT(5X'KS, CS, DT   :  '3F8.2/)
106   FORMAT(5X'INELASTIC CABLE',A)
107   FORMAT(5X'ELASTIC CABLE',A)
108   FORMAT(5X'NO AERODYNAMICS')
109   FORMAT(5X'DRAG ONLY AERODYNAMICS')
110   FORMAT(5X'CONEX AERODYNAMICS')
111   FORMAT(5X'WAKE MODEL NOT SELECTED')
112   FORMAT(5X'AXIAL WAKE MODEL SELECTED')
113   FORMAT(5X'3-D WAKE MODEL SELECTED')
114   FORMAT(5X'INPUT CONTROL HISTORY AUTOPILOT DISENGAGED')
115   FORMAT(5X'INPUT CONTROL HISTORY AUTOPILOT ENGAGED')
116   FORMAT(5X'LATERAL AXIS COMPUTER GENERATED FREQUENCY SWEEP')
117   FORMAT(5X'LONGITUDINAL AXIS COMPUTER GENERATED FREQUENCY SWEEP')
118   FORMAT(5X'COLLECTIVE AXIS COMPUTER GENERATED FREQUENCY SWEEP')
119   FORMAT(5X'DIRECTIONAL AXIS COMPUTER GENERATED FREQUENCY SWEEP')

14    WRITE(6,12)
12    FORMAT('!!! ERROR OPENING INPUT FILE ghsl.dat !!!')
      STOP

15    WRITE(6,13)
13    FORMAT('!!! ERROR OPENING INPUT FILE loadstab.dat !!!')
      STOP

      END
```

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX L NRT_UNC3_OUT.F

Output module for GenHel / Slung Load program. Writes formatted output to binary file for analysis. Load stability outputs were added. Program by Tyson, P. modified by Ehlers, G.

```
C file genhel/batch/sl/nrt_unc3_out.f  ----------------------  31 AUG
98 Peter Tyson
C small changes to output list jun 00

C     Writes formatted output to binary file for GenHel/Slung Load
C  simulation.  Output file 'name.dat' is then read and reformatted
C  by ghsl_dat.f (runghsldat) to create 'name.out' (ASCII header file)
C  and 'name.xp' (UNC3 format XPLOT/CIFER file).
C     This subroutine is called by BHAWK_NRT_IO in the non-real-time
C  run of GenHel, version 6.0sl.  NRESET true opens output files and
C  writes header information.  NREC contains the number of records read
C  from the input file (done after first time through NRT_UNC3_OUT
C  and so is written to data file on second pass)

      SUBROUTINE NRT_UNC3_OUT(NRESET)

      INCLUDE 'slvars.cmn'

      COMMON /ISCASC/ ISCAS(50)
      EQUIVALENCE (ISCAS(6), NGAJFPS)
      EQUIVALENCE (A(148), FNORTH), (A(149), FE),  (A(150), FD)
      EQUIVALENCE (A(164), TTL),    (A(165), TTM), (A(166), TTN)
      EQUIVALENCE (FCS(67), XAT), (FCS(68), XBT), (FCS(69), XPT)

      CHARACTER*40 FN, FN2
      INTEGER FIRSTPASS
      DATA FIRSTPASS /0/

C--- first pass after reset, open output file, do IC writes set
firstpass = 1 and return  ---C
C--- second pass write NREC, set firstpass = 0, and write record
C--- write record each cycle thereafter

      IF (NRESET .NE. 0) THEN
         NRESET = 0
         FIRSTPASS = 1
         NLAST = LASTCHR(DATAFILE)
         FN    = DATAFILE(1:NLAST)//".dat"
         OPEN(1,FILE=FN,FORM='UNFORMATTED',STATUS='UNKNOWN')
         WRITE(1) NSTORE, DT, ILOAD, IWAKE, ISWIRL, IPILOT, IDATA,
     X      NGAJFPS, TOW, FWT, W1, I1XX, I1YY, I1ZZ, I1XZ, AXIS, R1SN,
     X      PH1DEG, TH1DEG, PS1DEG, (A(J),J=64,66), PSVA, FA11, MA11,
     X      XAAD, XBAD, XCAD, XPAD
         WRITE(1) CHFILE
```

```
C                                                                no
load ic write
         IF (ILOAD.EQ.0) WRITE(6,17) FN
C                                                          slsc ic
write
         IF (ILOAD.EQ.1) THEN
            WRITE(6,18) FN
            WRITE(1) LOADNAME, W2, I2XX, I2YY, I2ZZ, I2XZ, R1SA1,
     X         R2S12, DOQ, STRETCH, LCO, LC, KS, CS, QO, V1S1,
     X         FA1N, FA2N, PH2*R2D, TH2*R2D, PS2*R2D, PHC, THC, Q(12),
     X         (U(j),j=1,12)
         END IF
C                                                          slmc ic
write
         IF (ILOAD.EQ.2) THEN
            WRITE(6,19) FN
            WRITE(1) LOADNAME, W2, I2XX, I2YY, I2ZZ, I2XZ, STRETCH,
     X         IAERSL, NC, R2SJ2, RA2PO2, R2P2SO2, R2P2S2, KS, CS,
     X         R1SA1, R2PJ2, DOQ, GAM*RTD, PSVA*RTD, ALF2D, BET2D,
     X         DETG, LCJO, LCJ, TAUJ, FA1N, FC11, FC1N, MC11, FA22,
     X         FA2N, MA22, (A(J),J=64,66), PH2*R2D, TH2*R2D, PS2*R2D,
     X         RA2S2, VA2SN, R1S2SN, DYNAMIC,OM22, DV2SN, KSWRL,
     X         KDYN,KR2,KPS2
         END IF
C                                                          wake ic
write
         IF ((ILOAD.NE.0).AND.(IWAKE.EQ.1)) THEN
            FN2 = DATAFILE(1:NLAST)//".da2"
            OPEN(3,FILE=FN2,FORM='UNFORMATTED',STATUS='UNKNOWN')
            WRITE(3) NSTORE
            WRITE(6,20) FN2
         END IF
         RETURN
      END IF

C--- write number of input data points on second pass only ---C
C    set NREC = LLLOAD(12).  Mystery - sometimes the code doesn't
equivalence nrec to lload(12)
C    as specified in slvars.cmn  Unpredictable occurrence.

      IF (FIRSTPASS.NE.0) THEN
         NREC = LLOAD(12)
         WRITE(1) NREC
         FIRSTPASS = 0
      END IF

C--- write data each time called ---C
C                                                          no load
      IF (ILOAD.EQ.0) WRITE(1) T, (A(j),j=88,90), (A(j),j=55,57),
     X   (A(j),j=64,66), (A(j),j=37,39), (A(j),j=4,9),
     X   A(106), A(107), A(176), FA11, MA11, XAAD, XBAD, XCAD, XPAD,
     X   RSAS, PSAS, YSAS, DMIXA, DMIXB, DMIXC, DMIXP, PSFWD, PSAFT,
     X   PSLAT, PSTR
C                                                          slsc
      IF (ILOAD.EQ.1) WRITE(1) T, DU, TAU, U, (DQ(I),I=4,11), Q,
     X   R2SN, V2SN, DV2SN, RSN, VSN, DELRSN, DELVSN, DELDVSN, SSFN,
     X   ISSFN, IISSFN, FA11, FA1N, FA2N, MA11, MA22, OMCC, LCMLCO,
```

```
     X   RHK2S1, FC11, MC11, XAAD, XBAD, XCAD, XPAD, PHCH, THCH,
     X   RPHC, RTHC, RSAS, PSAS, YSAS, DMIXA, DMIXB, DMIXC, DMIXP,
     X   PSFWD, PSAFT, PSLAT, PSTR
C                                                            slmc, long
list
      IF ((ILOAD.EQ.2).AND.(IDATA.EQ.1)) WRITE(1) T, DU, U,
     X   (DQ(I),I=4,9), Q, P2P, Q2P, FC11, FC1N, MC11, FA11, FA1N,
MA11,
     X   FA22, FA2N, MA22, FA2W, MA2W, V2SN, VA2S2, RA2S1, R1S2SN,
     X   XAAD, XBAD, XCAD, XPAD, RSAS, PSAS, YSAS, DMIXA, DMIXB, DMIXC,
     X   DMIXP, PSFWD, PSAFT, PSLAT, PSTR, float(QUAD), ALF2D, BET2D,
     X   DALF2D, DBET2D, DANG2D, P21, Q21, R21, P2N, Q2N, R2N,test,
     X   DELXAAD,DELXBAD,DELXPAD,DELXCAD,CAD_SAVE,AMGS2,DV1S1,
     X   F1V, F2V, F3V, F1H, F2H, F3H, SPEED
C                                                            slmc, short
list
      IF((ILOAD.EQ.2).AND.(IDATA.EQ.0)) WRITE(1) T, (U(I),I=4,6),
     X   XAAD, XBAD, XCAD, XPAD, P2P, Q2P, RSAS, PSAS, YSAS, DMIXA,
     X   DMIXB, DMIXC, DMIXP,DELXAAD,DELXBAD,DELXPAD,DELXCAD

      RETURN
   15 FORMAT(/)
   17 FORMAT(/5x'WRITING NO-LOAD HEADER TO '16A)
   18 FORMAT(/5x'WRITING SL SINGLE CABLE HEADER TO '16A)
   19 FORMAT(/5x'WRITING SL MULTI-CABLE HEADER TO '16A)
   20 FORMAT(/5x'WRITING WAKE INFORMATION TO '16A)
      END
```

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX M GHSLDAT

Data input module for GenHel / Slung Load simulation. Yaw degree of freedom correction parameters were added. Program by Tyson, P. modified by Ehlers, G.

```
C file /genhel/batch/ghsl.dat  -------------------------------  31 AUG
98 Peter Tyson
C Yaw DOF Corrections by George Ehlers July 01

C  Data input for GenHel/Slung Load simulation, read by GHSL_INIT.
C  To change case, comment &SLDATA line for all cases except desired.
 &HCDATA
     TOW      =   14689.0,
     XMOMTO   = 5307900.0,
     FWT      =    2000.0,
     FWMX     =    2360.0,
     AIRSPEED =        60,

 &END
 &SLRUN
     DATAFILE ='junk'
     CHFILE   ='flt169.18ghch'
     AXIS     =  1,
     DYNAMIC  =  0,
     STRETCH  =  1,
     IAERSL   =  4,
     ILOAD    =  2,
     IPILOT   =  1,
     IWAKE    =  0,
     ISWIRL   =  0,
     IDATA    =  1,
     STABSYST =  1,

 &END

C    CHFILE   =  '/ctr/cicolani/flt172.12ch',
C    AIRSPEED =       0.0,

C Helicopter Data:
C  TOW        Take Off Weight
C  XMOMTO     X- Moment Arm based on takeoff weight (with full fuel
load)
C  FWT        Fuel Weight at time of maneuver to be simulated
C  FWMX       Maximum Fuel weight at takeoff

C Run Options:
c  DATAFILE   filename for output file fn.dat (fn.out, fn.xp are
generated by ghsl_dat.f)
c  CHFILE     filename for control history inputs. binary files read in
nrt_unc3_in.f
```

```
              can be sweep.dat from makesweep or fltxxx_yy.dat flight
control histories
C  AXIS       (1) Lat  (2) Long  (3) Coll  (4) Dir  (only req'd for
sweep.dat)
C  STRETCH    (0) inelastic cables  (1) elastic cables
C  IAERSL     (1) drag only  (2) CONEX static aero (PT/techdat3), (3)
conex4 (LC/techdat4)
C  ILOAD      (0) no-load    (1) SLSC (1K Plate load)   (2) SLMC
C  IPILOT     (0) no feedback into control inputs  (1) feedback
included
C  IWAKE      (0) no wake model  (1) wake model incorporated
C  ISWIRL     (0) no swirl in wake  (1) swirl in wake (tangential
velocity)
C  IDATA      (0) minimum data for CIFER analysis  (1) full data output

C UH60 flight test case (ballasted CONEX with instrumentation and
C standard 4-cable sling).  LIFT PT coords revised 14feb97,
C Inertia and coords adjusted 13may97, 14jul98.
C Lift points, r2pj2, are set away from edge of conex a little.
C R2SS2 = load-cg-to-accelerometer vector.  use zero if no load
accelerometers

 &SLDATA
     LOADNAME = '4K CONEX (BALLASTED) WITH INST PKG      ',
     W2       =    4105.0,
     I2XX     =    1876.0,
     I2YY     =    1482.2,
     I2ZZ     =    1376.0,
     I2XZ     =       0.0,
     KS       =    9645.0,
     CS       =      22.0,
     RA2PO2   =   0.0, 0.0, 18.3036,
     R2P2SO2  =   0.0, 0.0,  1.38,
     R2P2S2   =   0.0, 0.0,  1.38,
     NC       =   4,
     R2PJ2    =   2.8073, -4.0626, -3.2032,
                  2.8073,  4.0626, -3.2032,
                 -2.8073, -4.0626, -3.2032,
                 -2.8073,  4.0626, -3.2032, 12*0,
     DOQ      =  0,
     DELPS2O  =  -45,
     R2OD     =  100,
     R2SS2    =  1.031, .575, -.899,
     KSWRL    = 40
     KR2      = 1.5
     KDYN     = .3
 &END

C empty CONEX load, including instrumentation.

C&SLDATA
     LOADNAME = '2K CONEX (UNBALLASTED) WITH INST PKG    ',
     W2       =    1794.0,
     I2XX     =     785.3,
     I2YY     =     590.7,
     I2ZZ     =     766.0,
     I2XZ     =       0.0,
```

```
     KS        =    9645.0,
     CS        =      22.0,
     RA2PO2    =   0.0, 0.0, 18.3036,
     R2P2SO2   =   0.0, 0.0,  1.38,
     R2P2S2    =   0.0, 0.0,  1.38,
     NC        =   4
     R2PJ2     =   2.8073, -4.0626, -3.2032,
                   2.8073,  4.0626, -3.2032,
                  -2.8073, -4.0626, -3.2032,
                  -2.8073,  4.0626, -3.2032, 12*0,
     DOQ       =  50.0,
     R2OD      =  0
 &END

     R2SS2     = .891, .575, .276,

C 4154lb 35.5 x 35.5 x 14.625 in steel block UH60 97 flt test load -
1997 tests.
C Adjusted 14jul98.
C Lift point coords, r2pj2, are set away from edge of block a little.
C ra2po2 changed from 0,0,18.14863 to 0,0,16.4391, 20 jul 98 pht

C&SLDATA
     LOADNAME = '4K STEEL BLOCK (no instrumentation)',
     W2        =    4154.0,
     I2XX      =     109.6,
     I2YY      =     109.6,
     I2ZZ      =     187.1,
     I2XZ      =       0.0,
     KS        =    9645.0,
     CS        =      22.0,
     RA2PO2    =   0.0, 0.0, 16.4391,
     R2P2SO2   =   0.0, 0.0,  0.0,
     R2P2S2    =   0.0, 0.0,  0.0,
     NC        =   4
     R2PJ2     =   1.32, -1.32, -0.61,
                   1.32,  1.32, -0.61,
                  -1.32, -1.32, -0.61,
                  -1.32,  1.32, -0.61, 12*0,
     DOQ       =  10.0,
     R2SS2     = 0, 0, 0,
 &END

C 4K steel block load - 1999 flight tests, including instrumentation
C&SLDATA
     LOADNAME = '4K STEEL BLOCK',
     W2        =    3895.0,
     I2XX      =     103.7,
     I2YY      =     104.6,
     I2ZZ      =     174.3
     I2XZ      =       0.0,
     KS        =    9645.0,
     CS        =      22.0,
     RA2PO2    =   0.0, 0.0, 16.4086,
     R2P2SO2   =   0.0, 0.0,  0.0,
     R2P2S2    =   0.0, 0.0,  0.0,
     NC        =   4
```

```
     R2PJ2    =   1.27, -1.27, -0.61,
                  1.27,  1.27, -0.61,
                 -1.27, -1.27, -0.61,
                 -1.27,  1.27, -0.61, 12*0,
     DOQ      =  10.0,
     R2SS2    = .787, .244, -.688,
 &END



C 6384lb steel block, weighed 28 jul 98 on the S/R (Bldg 255) 10K
pallet
C scale.  Some water was in the block (leaked out when lifted)
C unknown what total water capacity is...
C Parameters calculated 15 jul 98, recalc 28 jul 98
c&SLDATA
     LOADNAME = '6K STEEL BLOCK                          ',
     W2       =    6384.0,
     I2XX     =     308.4,
     I2YY     =     296.2,
     I2ZZ     =     470.7,
     I2XZ     =       0.0,
     KS       =    9645.0,
     CS       =      22.0,
     RA2PO2   =   0.0,  0.0,  16.4741,
     R2P2SO2  =   0.0, -0.118 -0.179,
     R2P2S2   =   0.0, -0.118 -0.179,
     NC       =   4,
     R2PJ2    =   1.8333, -1.8333, -0.6792,
                  1.8333,  1.8333, -0.6792,
                 -1.8333, -1.8333, -0.6792,
                 -1.8333,  1.8333, -0.6792, 12*0,
     DOQ      =   0.0,
     R2SS2    = 0, 0, 0,
 &END

C 1130lb steel plate, use with ILOAD = 1 (SLSC) option
c&SLDATA
     LOADNAME = '1K STEEL PLATE                          ',
     W2       =    1130.0,
     I2XX     =       9.1,
     I2YY     =     119.3,
     I2ZZ     =     128.1,
     I2XZ     =       0.0,
     KS       =    4172.3,
     CS       =       0.0,
     LCO      =      22.7083,
     R2S12    =   0.0, 0.0, -8.36,
     DOQ      =      10.0,
     R2SS2    =  0, 0, 0,
 &END
C (DOQ normally 10.0)
```

# APPENDIX N GHSL_DAT.F

Writes binary data file into unc3 format for plotting. Load stabilization data was added. Program by Tyson, P. modified by Ehlers, G.

```
C  file /u2/harrier/tyson/GenHel/batch/ghsl_dat.f -------------------
31 AUG 98, Peter Tyson

C  Data print and plot storage file for GenHel/Slung Load simulation.
C  Parameters nchanx = number of variables stored in dat array for (1)
no-load, (2) slsc, and (3) slmc cases.
C  ILOAD = (0,1,2) -> (no load, slsc, slmc) variable list.  IDATA =
(0,1) -> short,long list for slmc.

      PROGRAM GHSL_DAT

      PARAMETER(NCHAN1 =  43)
      PARAMETER(NCHAN2 = 126)
      PARAMETER(NCHAN3 = 137)
      PARAMETER(NCHAN4 =  21)
      PARAMETER(NCHANW =  53)
      PARAMETER(NCHANX =  21)

      REAL FA11(3), FA1N(3), MA11(3), FC11(3), FC1N(3), MC11(3),
     X     FA22(3), FA2N(3), MA22(3), FC22(3), FC2N(3), MC22(3),
     X     R1SN(3), V1SN(3), R1SA1(3), VA2SN(3), R2SJ2(24), RA2S2(3),
     X     RA2PO2(3),
     X     R2P2SO2(3), R2P2S2(3), R2PJ2(3,8), R2S12(3), V1S1(3),
     X     QO(12), Q(12), U(12), LCO, LC, KS, I1XX, I1YY, I1ZZ, I1XZ,
     X     I2XX, I2YY, I2ZZ, I2XZ, C(4), TAUJ(8), LCJ(8), LCJO(8),
     X     DHOOK, HHOOK, DAT1(NCHAN1), DAT2(NCHAN2), DAT3(NCHAN3),
     X     DAT4(NCHAN4), DATW(NCHANW), DATX(NCHANX), DALF2D, DBET2D,
     X     DANG2D, ALF2D, BET2D, R1S2SN(3), DYNAMIC, P21, Q21, R21,
     X     AMGS2(3),OM22(3),DV2SN(3),
     X     KSWRL,KDYN,KR2,KPS2,KTSTFPS

      REAL*8 TIME, DAT81(NCHAN1), DAT82(NCHAN2), DAT83(NCHAN3+1),
     X       DAT84(NCHAN4), DAT8W(NCHANW), DAT8X(NCHANX)

      INTEGER STRETCH, JRTD1(12), JRTD2(31), JRTD3(32), JRTD4(5),
     X        NREC, AXIS
      INTEGER*4 UNIT
      LOGICAL*4 openW, L

      CHARACTER NAME*30, FIN*30, FOUT*30, FXP*30, SN1(NCHAN1)*16,
     X          SN2(NCHAN2)*16, SN3(NCHAN3+1)*16, SN4(NCHAN4)*16,
     X          SNW(NCHANW)*16, SNX(NCHANX)*16, LOADNAME*40, CABLES,
     X          CHFILE*40

      DATA UNIT/3/,
     X RTD, G, KTSTFPS/57.2957795, 32.174, 1.687810/,
```

```
     X JRTD1/5,6,7,11,12,13,14,15,16,17,18,19/,
     X JRTD2/5,6,7,8,9,10,18,19,20,21,22,23,27,28,29,30,31,32,
     X       33,34,38,39,40,41,42,43,44,45,95,96,97/,
     X JRTD3/5,6,7,8,9,10,17,18,19,20,21,22,26,27,28,29,30,31,
     X       35,36,37,38,39,40,44,45,112,113,114, 115, 116, 117/,
     X JRTD4/2,3,4,9,10/
C
NO LOAD SIGNAL LIST
      DATA SN1/
     X  'T1'      ,'dv1snx'  ,'dv1sny'  ,'dv1snz'  ,'dp1'     ,
     X  'dq1'     ,'dr1'     ,'v1snx'   ,'v1sny'   ,'v1snz'   ,
     X  'p1'      ,'q1'      ,'r1'      ,'ph1'     ,'th1'     ,
     X  'ps1'     ,'dph1'    ,'dth1'    ,'dps1'    ,'r1snx'   ,
     X  'r1sny'   ,'r1snz'   ,'fa11x'   ,'fa11y'   ,'fa11z'   ,
     X  'LA1'     ,'MA1'     ,'NA1'     ,'da'      ,'db'      ,
     X  'dc'      ,'dp'      ,'RSAS'    ,'PSAS'    ,'YSAS'    ,
     X  'DMIXA'   ,'DMIXB'   ,'DMIXC'   ,'DMIXP'   ,'PSFWD'   ,
     X  'PSAFT'   ,'PSLAT'   ,'PSTR'    /
C
SLSC SIGNAL LIST
      DATA SN2/
     X  'T'       ,'dv1snx'  ,'dv1sny'  ,'dv1snz'  ,'dp1'     ,
     X  'dq1'     ,'dr1'     ,'dp2'     ,'dq2'     ,'dr2'     ,
     X  'dva1cx'  ,'dva1cy'  ,'ddlc'    ,'tau/w1'  ,'v1snx'   ,
     X  'v1sny'   ,'v1snz'   ,'p1'      ,'q1'      ,'r1'      ,
     X  'p2'      ,'q2'      ,'r2'      ,'va1cx'   ,'va1cy'   ,
     X  'dlc'     ,'dph1'    ,'dth1'    ,'dps1'    ,'dph2'    ,
     X  'dth2'    ,'dps2'    ,'dphc'    ,'dthc'    ,'r1snx'   ,
     X  'r1sny'   ,'r1snz'   ,'ph1'     ,'th1'     ,'ps1'     ,
     X  'ph2'     ,'th2'     ,'ps2'     ,'phc'     ,'thc'     ,
     X  'lc'      ,'r2snx'   ,'r2sny'   ,'r2snz'   ,'v2snx'   ,
     X  'v2snx'   ,'v2snz'   ,'dv2snx'  ,'dv2sny'  ,'dv2snz'  ,
     X  'rsnx'    ,'rsny'    ,'rsnz'    ,'vsnx'    ,'vsny'    ,
     X  'vsnz'    ,'delrsx'  ,'delrsy'  ,'delrsz'  ,'delvsx'  ,
     X  'delvsy'  ,'delvsz'  ,'dldvsx'  ,'dldvsy'  ,'dldvsz'  ,
     X  'ssfnx'   ,'ssfny'   ,'ssfnz'   ,'issfnx'  ,'issfny'  ,
     X  'issfnz'  ,'iissfx'  ,'iissfy'  ,'iissfz'  ,'fa11x'   ,
     X  'fa11y'   ,'fa11z'   ,'fa1nx'   ,'fa1ny'   ,'fa1nz'   ,
     X  'fa2nx'   ,'fa2ny'   ,'fa2nz'   ,'ma11x'   ,'ma11y'   ,
     X  'ma11z'   ,'ma22x'   ,'ma22y'   ,'ma22z'   ,'pc'      ,
     X  'qc'      ,'rc'      ,'lcmlco'  ,'rhk2s1x' ,'rhk2s1y' ,
     X  'rhk2s1z' ,'fc11x'   ,'fc11y'   ,'fc11z'   ,'LC1'     ,
     X  'MC1'     ,'NC1'     ,'da'      ,'db'      ,'dc'      ,
     X  'dp'      ,'phch'    ,'thch'    ,'rphc'    ,'rthc'    ,
     X  'RSAS'    ,'PSAS'    ,'YSAS'    ,'DMIXA'   ,'DMIXB'   ,
     X  'DMIXC'   ,'DMIXP'   ,'PSFWD'   ,'PSAFT'   ,'PSLAT'   ,
     X  'PSTR'    /
C
SLMC: LONG LIST
      DATA SN3/
     X  'T1'      ,'dv1snx'  ,'dv1sny'  ,'dv1snz'  ,'dp1'     ,
     X  'dq1'     ,'dr1'     ,'dp2'     ,'dq2'     ,'dr2'     ,
     X  'ddra2s2x','ddra2s2y','ddra2s2z','v1snx'   ,'v1sny'   ,
     X  'v1snz'   ,'p1'      ,'q1'      ,'r1'      ,'p2'      ,
     X  'q2'      ,'r2'      ,'dra2s2x' ,'dra2s2y' ,'dra2s2z' ,
     X  'dph1'    ,'dth1'    ,'dps1'    ,'dph2'    ,'dth2'    ,
     X  'dps2'    ,'r1snx'   ,'r1sny'   ,'r1snz'   ,'ph1'     ,
```

```
     X  'th1'     ,'ps1'     ,'ph2'     ,'th2'     ,'ps2'      ,
     X  'ra2s2x'  ,'ra2s2y'  ,'ra2s2z'  ,'p2p'     ,'q2p'      ,
     X  'fc11x'   ,'fc11y'   ,'fc11z'   ,'fc1nx'   ,'fc1ny'    ,
     X  'fc1nz'   ,'LC1'     ,'MC1'     ,'NC1'     ,'fa11x'    ,
     X  'fa11y'   ,'fa11z'   ,'fa1nx'   ,'fa1ny'   ,'fa1nz'    ,
     X  'LA1'     ,'MA1'     ,'NA1'     ,'fa22x'   ,'fa22y'    ,
     X  'fa22z'   ,'fa2nx'   ,'fa2ny'   ,'fa2nz'   ,'LA2'      ,
     X  'MA2'     ,'NA2'     ,'fa2wx'   ,'fa2wy'   ,'fa2wz'    ,
     X  'LA2W'    ,'MA2W'    ,'NA2W'    ,'v2snx'   ,'v2sny'    ,
     X  'v2snz'   ,'va2s2x'  ,'va2s2y'  ,'va2s2z'  ,'ra2s1x'   ,
     X  'ra2s1y'  ,'ra2s1z'  ,'r1s2snx' ,'r1s2sny' ,'r1s2snz'  ,
     X  'da'      ,'db'      ,'dc'      ,'dp'      ,'RSAS'     ,
     X  'PSAS'    ,'YSAS'    ,'DMIXA'   ,'DMIXB'   ,'DMIXC'    ,
     X  'DMIXP'   ,'PSFWD'   ,'PSAFT'   ,'PSLAT'   ,'PSTR'     ,
     X  'quad'    ,'alf2d'   ,'bet2d'   ,'dalf2d'  ,'dbet2d'   ,
     X  'dang2d'  ,'p21'     ,'q21'     ,'r21'     ,'p2n'      ,
     X  'q2n'     ,'r2n'     ,
     X  'delxaad' ,'delxbad' ,'delxpad' ,'delxcad' ,'xaad_sav',
     X  'xbad_sav','xpad_sav','xcad_sav','AMGS2X'  ,'AMGS2Y'   ,
     X  'AMGS2Z'  ,'dv1s1x'  ,'dv1s1y'  ,'dv1s1z'  ,
     X  'fxv'     ,'fyv'     ,'fzv'     ,
     X  'fxh'     ,'fyh'     ,'fzh'     ,'speed'/

C
SLMC: SHORT LIST
      DATA SN4/
     X  'T1'      ,'p1'      ,'q1'      ,'r1'      ,'da'       ,
     X  'db'      ,'dc'      ,'dp'      ,'p2p'     ,'q2p'      ,
     X  'RSAS'    ,'PSAS'    ,'YSAS'    ,'DMIXA'   ,'DMIXB'    ,
     X  'DMIXC'   ,'DMIXP'   ,
     X  'delxaad' ,'delxbad' ,'delxpad' ,'delxcad'/
C
WAKE: LONG LIST
      DATA SNW/
     X  'T'       ,'a1f'     ,'b1f'     ,'v1x'     ,'v1y'      ,
     X  'v1z'     ,'vo'      ,'vo1x'    ,'vo1y'    ,'vo1z'     ,
     X  'vpp1x'   ,'vpp1y'   ,'vpp1z'   ,'vppnx'   ,'vppny'    ,
     X  'vppnz'   ,'vpp'     ,'vp1x'    ,'vp1y'    ,'vp1z'     ,
     X  'vpnx'    ,'vpny'    ,'vpnz'    ,'vp'      ,'phiw'     ,
     X  'thew'    ,'xcw'     ,'ycw'     ,'rh2swx'  ,'rh2swy'   ,
     X  'rh2swz'  ,'rh2swox' ,'rh2swoy' ,'rh2swoz' ,'thetap'   ,
     X  'radius'  ,'rad'     ,'height'  ,'vwz'     ,'vwt'      ,
     X  'pst'     ,'vwwx'    ,'vwwy'    ,'vwwz'    ,'vw2x'     ,
     X  'vw2y'    ,'vw2z'    ,'va2s2wx' ,'va2s2wy' ,'va2s2wz'  ,
     X  'va2s2x'  ,'va2s2y'  ,'va2s2z'  /
C
WAKE: SHORT LIST
      DATA SNX/
     X  'T'       ,'a1f'     ,'b1f'     ,'vo'      ,'vpp'      ,
     X  'vp'      ,'phiw'    ,'thew'    ,'xcw'     ,'ycw'      ,
     X  'rh2swox' ,'rh2swoy' ,'rh2swoz' ,'thetap'  ,'radius'   ,
     X  'height'  ,'vwz'     ,'pst'     ,'va2s2wx' ,'va2s2wy'  ,
     X  'va2s2wz' /


 10   TYPE *, 'Enter name of input file name.dat'
      READ(5,11,ERR=10) NAME
```

```
      NLAST = LASTCHR(NAME)
      FIN  = NAME(1:nlast)//".dat"
      FOUT = NAME(1:nlast)//".out"
      FXP  = NAME(1:nlast)//".xp"
      WRITE(6,11) FIN
 11   FORMAT(A)

      OPEN(1,FILE=FIN,FORM='unformatted',STATUS='old')

      READ(1) NS, DT, ILOAD, IWAKE, ISWIRL, IPILOT, IDATA, NGAJFPS,
     X   TOW, FWT, W1, I1XX, I1YY, I1ZZ, I1XZ, AXIS, R1SN,
     X   PH1DEG, TH1DEG, PS1DEG, V1SN, PSVA, FA11, MA11, C
      READ(1) CHFILE

      OPEN(2,FILE=FOUT,FORM='formatted',STATUS='unknown')
      WRITE(2,151) FOUT

      IF (ILOAD.EQ.0) L = openW(UNIT,FXP,NCHAN1,SN1,'unc3')
      IF (ILOAD.EQ.1) L = openW(UNIT,FXP,NCHAN2,SN2,'unc3')
      IF ((ILOAD.EQ.2).AND.(IDATA.EQ.1))
     X   L = openW(UNIT,FXP,NCHAN3+1,SN3,'unc3')
      IF ((ILOAD.EQ.2).AND.(IDATA.EQ.0))
     X   L = openW(UNIT,FXP,NCHAN4,SN4,'unc3')
      IF (.NOT.L) STOP'openW'

      IF (ILOAD.EQ.1) THEN
         READ(1) LOADNAME, W2, I2XX, I2YY, I2ZZ, I2XZ, R1SA1, R2S12,
     X      DOQ, STRETCH, LCO, LC, KS, CS, QO, V1S1, FA1N, FA2N,
     X      R1SN, PH2, TH2, PS2, PHC, THC, LC, (U(j),j=1,12)
         CABLES = ' '
         WRITE(2,101)
      END IF

      IF (ILOAD.EQ.2) THEN
         READ(1) LOADNAME, W2, I2XX, I2YY, I2ZZ, I2XZ, STRETCH,
     X      IAERSL, NC, R2SJ2, RA2PO2, R2P2SO2, R2P2S2, KS, CS,
     X      R1SA1, R2PJ2, DOQ, GAMA, PSVA, ALF2D, BET2D, DETG, LCJO,
     X      LCJ, TAUJ, FA1N, FC11, FC1N, MC11, FA22, FA2N, MA22, V1SN,
     X      PH2DEG, TH2DEG, PS2DEG, RA2S2, VA2SN, R1S2SN, DYNAMIC,
     X      OM22,DV2SN,KSWRL,KDYN,KR2,KPS2
         CABLES = 'S'
         WRITE(2,102)
      END IF

      READ(1) NREC
      RUNTIME = NREC * DT

      IF (CHFILE.EQ.'sweep.dat') THEN
         IF (AXIS.EQ.1) THEN
            WRITE(2,103)
         ELSE IF (AXIS.EQ.2) THEN
            WRITE(2,104)
         ELSE IF (AXIS.EQ.3) THEN
            WRITE(2,105)
         ELSE
            WRITE(2,106)
```

```
         END IF
      ELSE
         WRITE(2,107) CHFILE
      END IF

      IF (ILOAD.EQ.0) WRITE(2,100)

      WRITE(2,152) NREC, DT, I1XX, I1YY, I1ZZ, I1XZ
      WRITE(2,114) TOW, FWT, W1

      IF (ILOAD.EQ.0) THEN
         WRITE(2,153) R1SN, PH1DEG, TH1DEG, PS1DEG, V1SN, PSVA,
     X      FA11, MA11, C
      ELSE
         WRITE(2,115) W1+W2
         WRITE(2,112) LOADNAME, W2, I2XX, I2YY, I2ZZ, I2XZ
         IF (ILOAD.EQ.2) WRITE(2,113) RA2PO2, R2P2SO2, R2P2S2,
     X      (R2PJ2(1,J),R2PJ2(2,J),R2PJ2(3,J),J=1,NC),R1S2SN

         IF (STRETCH.EQ.0) WRITE(2,116) CABLES
         IF (STRETCH.EQ.1) WRITE(2,117) CABLES

         IF ((IAERSL.EQ.0).OR.((IAERSL.EQ.1).AND.(DOQ.EQ.0.))) THEN
            WRITE(2,118)
         ELSE
            IF (IAERSL.EQ.1) WRITE(2,119)
            IF (IAERSL.EQ.2) WRITE(2,120) DYNAMIC
            IF (IAERSL.EQ.3) WRITE(2,1201)
            IF (IAERSL.EQ.4) WRITE(2,1202)
            IF (IWAKE.EQ.0) WRITE(2,121)
            IF (IWAKE.EQ.1) THEN
               IF (ISWIRL.EQ.0) WRITE(2,122)
               IF (ISWIRL.EQ.1) WRITE(2,123)
            END IF
         END IF

      END IF

      IF (NGAJFPS.EQ.0) WRITE(2,124)
      IF (NGAJFPS.EQ.1) WRITE(2,125)

      IF (IPILOT.EQ.0) WRITE(2,126)
      IF (IPILOT.EQ.1) WRITE(2,127)

      IF (ILOAD.EQ.1) THEN
         WRITE(2,201) R1SA1, R2S12, LCO
         IF (STRETCH.EQ.1) WRITE(2,202) KS, CS
         WRITE(2,203) R1SN, PH1DEG, TH1DEG, PS1DEG, PH2, TH2, PS2,
     X      PHC, THC, LC, V1S1, FA1N, FA2N, MA11, (Q(j),U(j),j=1,12)
      END IF

      IF (ILOAD.EQ.2) THEN
         WRITE(2,301) R1SA1
         IF (STRETCH.NE.0) WRITE(2,302) KS, CS, NC, DETG,
     X      (TAUJ(j),LCJ(j),LCJO(j),j=1,NC)
         WRITE(2,303) KSWRL,KDYN,KR2,KPS2
         THT = ATAN(SQRT(FA2N(1)**2+FA2N(2)**2)/(W2+FA2N(3)))*RTD
```

```
          R1SN(3) = -R1SN(3)
          WRITE(2,305) R1SN,V1SN,PH1DEG,TH1DEG,PS1DEG,PH2DEG,TH2DEG,
     X      PS2DEG,OM22,DV2SN,RA2S2,VA2SN,FA11,FA1N,MA11,FC11,FC1N,MC11,
     X      FA22,FA2N,MA22,ALF2D,BET2D,THT,C
       END IF

       CLOSE(2)

C  SECTION 500.  STORE XPLOT FILE
C  read time histories.  Convert angles to degrees, change signs of y,z
position
C  coordinates for r1sn, r2sn, rsn. Note: DAT8n, n = 1,..,4 arrays are
double precision

       IF (ILOAD.EQ.0) THEN
          DO 511 i = 1,NREC-1
          READ(1,err=590) DAT1
          TIME = DAT1(1)
          DO 512 j = 1,12
          k = JRTD1(j)
 512      DAT1(k) = DAT1(k)*RTD
          DO 513 j = 1,NCHAN1
 513      DAT81(j) = DAT1(j)
 511      CALL fWrite(UNIT, TIME, DAT81)
          CALL closeW(UNIT)
          WRITE(6,580) FOUT, RUNTIME, FXP
          STOP
       END IF
       IF (ILOAD.EQ.1) THEN
          DO 521 i = 1,NREC-1
          READ(1,err=590) DAT2
          TIME = DAT2(1)
          DAT2(14) = DAT2(14)/W1
          DO 522 j = 1,31
          k = JRTD2(j)
 522      DAT2(k)  = DAT2(k)*RTD
          DO 523 k = 1,2
          DAT2(35+k) = - DAT2(35+k)
          DAT2(47+k) = - DAT2(47+k)
 523      DAT2(56+k) = - DAT2(56+k)
          DO 524 j = 1,NCHAN2
 524      DAT82(j) = DAT2(j)
 521      CALL fWrite(UNIT, TIME, DAT82)
          CALL closeW(UNIT)
          WRITE(6,580) FOUT, RUNTIME, FXP
       END IF
       IF ((ILOAD.EQ.2).AND.(IDATA.EQ.1)) THEN
          DO 531 i = 1,NREC-1
          READ(1,err=590) DAT3
          TIME = DAT3(1)
          DO 532 j = 1,32
          k = JRTD3(j)
 532      DAT3(k) = DAT3(k)*RTD
          DAT3(34) = - DAT3(34)
C  correct the load yaw angle to within +/- 180 deg for viewing in
xplot
C534      IF (DAT3(40).GT. 180.) DAT3(40) = DAT3(40) - 360
```

```
C        IF (DAT3(40).LT.-180.) DAT3(40) = DAT3(40) + 360
C        IF (DAT3(40).GT. 180.) GO TO 534
C        IF (DAT3(40).LT.-180.) GO TO 534
         DO 533 j = 1,NCHAN3
 533     DAT83(j) = DAT3(j)
         DAT83(NCHAN3+1) =
     x       SQRT(DAT3(14)**2+DAT3(15)**2+DAT3(16)**2)/KTSTFPS

 531     CALL fWrite(UNIT, TIME, DAT83)
         CALL closeW(UNIT)
         WRITE(6,580) FOUT, RUNTIME, FXP
      END IF
      IF ((ILOAD.EQ.2).AND.(IDATA.EQ.0)) THEN
         DO 561 i = 1,NREC-1
         READ(1,err=590) DAT4
         TIME = DAT4(1)
         DO 562 j = 1,5
         k = JRTD4(j)
 562     DAT4(k) = DAT4(k)*RTD
         DO 563 j = 1,NCHAN4
 563     DAT84(j) = DAT4(j)
 561     CALL fWrite(UNIT, TIME, DAT84)
         CALL closeW(UNIT)
         WRITE(6,580) FOUT, RUNTIME, FXP
      END IF
 599  CONTINUE
      IF (IWAKE.EQ.1) THEN
         FIN  = NAME(1:nlast)//".da2"
         FXP  = NAME(1:nlast)//".wake.xp"
         WRITE(6,11) FIN
         OPEN(4,FILE=FIN,FORM='unformatted',STATUS='old')

C  IDATA option removed for dynamic aero troubleshooting

         L = openW(UNIT,FXP,NCHANX,SNX,'unc3')
         IF (.NOT.L) STOP'openW'
         READ(4) junk
         DO 600 i = 1,NREC-1
            READ(4,err=592) DATX
            TIME = DATX(1)
            DO 651 j = 1,NCHANX
 651        DAT8X(j) = DATX(j)
            CALL fWrite(UNIT, TIME, DAT8X)
 600     CONTINUE
         CALL closeW(UNIT)
         WRITE(6,680) FXP
      END IF
      STOP

 151  FORMAT('DATA OUTPUT FILE ',16A//)
 100  FORMAT('NO LOAD SIMULATION')
 101  FORMAT('SINGLE LOAD, SINGLE CABLE SIMULATION')
 102  FORMAT('SINGLE LOAD, MULTI-CABLE SIMULATION')
 103  FORMAT('COMPUTER GENERATED LATERAL INPUT SWEEP'/)
 104  FORMAT('COMPUTER GENERATED LONGITUDINAL INPUT SWEEP'/)
 105  FORMAT('COMPUTER GENERATED DIRECTIONAL INPUT SWEEP'/)
 106  FORMAT('COMPUTER GENERATED COLLECTIVE INPUT SWEEP'/)
```

```
107  FORMAT('CONTROL INPUT FROM '40A//)
112  FORMAT(5X'LOAD DESCRIPTION :  'A/
    X       5X'LOAD WEIGHT      : 'F8.2,' LB'/
    X       5X'LOAD INERTIA XX  : 'F8.2,
    X       3X'LOAD INERTIA YY  : 'F8.2,' LB-FT-S**2'/
    X       5X'LOAD INERTIA ZZ  : 'F8.2,
    X       3X'LOAD INERTIA XZ  : 'F8.2,' LB-FT-S**2')
113  FORMAT(5X'RA2PO2   :  '3F8.2/
    X       5X'R2P2SO2  :  '3F8.2/
    X       5X'R2P2S2   :  '3F8.2/
    X       5X'R2PJ2    :  '8(3F8.2/17X)/
    X       5X'R1S2SN   :  '3F8.2)
114  FORMAT(5X'HELICOPTER TAKEOFF WEIGHT :  'F10.2,'  LB'/
    X       5X'           FUEL WEIGHT    :  'F10.2,'  LB'/
    X       5X'           CURRENT WEIGHT :  'F10.2,'  LB'/)
115  FORMAT(5X'TOTAL HELICOPTER AND LOAD :  'F10.2,'  LB'/)
116  FORMAT(5X'INELASTIC CABLE',A)
117  FORMAT(5X'ELASTIC CABLE',A)
118  FORMAT(5X'NO LOAD AERODYNAMICS')
119  FORMAT(5X'DRAG ONLY AERODYNAMICS')
120  FORMAT(5X'CONEX (techdat3) STATIC AERODYNAMICS'/
    X       5X'   DYNAMIC AERODYNAMIC COEFFICIENT = ',F4.1/)
1201 FORMAT(5X'CONEX4 (techdat4) STATIC AERODYNAMICS (|beta| < 90)'/)
1202 FORMAT(5X'CONEX4A(techdat4) STATIC AERODYNAMICS (|alfa| < 90)'/)
121  FORMAT(5X'WAKE MODEL NOT SELECTED')
122  FORMAT(5X'WAKE MODEL WITHOUT SWIRL SELECTED')
123  FORMAT(5X'3-D WAKE MODEL SELECTED')
124  FORMAT(5X'FLIGHT PATH STABILIZATION (FPS) DISENGAGED')
125  FORMAT(5X'FLIGHT PATH STABILIZATION (FPS) ENGAGED')
126  FORMAT(5X'INPUT CONTROL HISTORY AUTOPILOT DISENGAGED')
127  FORMAT(5X'INPUT CONTROL HISTORY AUTOPILOT ENGAGED')

152  FORMAT(5X'NREC, DT       :    'I8,F10.2/
    X       5X'HC INERTIA XX  :    'F8.2,
    X       3X'HC INERTIA YY  :    'F8.2,' LB-FT-S**2',/
    X       5X'HC INERTIA ZZ  :    'F8.2,
    X       3X'HC INERTIA XZ  :    'F8.2,' LB-FT-S**2',//)
153  FORMAT('TRIM'/
    X       5X'R1SN (x,y,z)          ', 3F10.1/
    X       5X'PH1, TH1, PS1 (DEG)   ', 3F10.2/
    X       5X'V1SN (x,y,z)          ', 3F10.2/
    X       5X'PSVA                  ', 1F10.2/
    X       5X'FA11 (x,y,z)          ', 3F10.1/
    X       5X'MA11 (l,m,n)          ', 3F10.1/
    X       5X'XAAD, XBAD, XCAD, XPAD', 4F10.2/)

201  FORMAT(5X'R1SA1              ' 3f10.2/
    X       5X'R2S12              ' 3f10.2/
    X       5X'LCO                ' f10.4/)
202  FORMAT(5X'KS, CS             ' 2f10.1/)
203  FORMAT('TRIM'/
    X       5X'R1SN                 '3F9.1/
    X       5X'PH1, TH1, PS1 (deg)', 3F9.2/
    X       5X'PH2, TH2, PS2 (deg)', 3F9.2/
    X       5X'PHC, THC, LC       ', 2F9.2, F9.4/
    X       5X'VREFN              ', 3F9.2/
    X       5X'FA1N               ', 3F9.1/
```

```
     X       5X'FA2N               ', 3F9.1/
     X       5X'MA11               ', 3F9.1//
     X       'INITIAL STATES, INCLUDING INITIAL OFFSETS FROM TRIM'/
     X       5X'         q - ft,deg        u - fps,rps '/
     X       5X'R1SNX', f15.6, 7x, f10.6/
     X       5X'R1SNY', f15.6, 7x, f10.6/
     X       5X'R1SNZ', f15.6, 7x, f10.6/
     X       5X'PH1  ', f15.6, 7x, f10.6/
     X       5X'TH1  ', f15.6, 7x, f10.6/
     X       5X'PS1  ', f15.6, 7x, f10.6/
     X       5X'PH2  ', f15.6, 7x, f10.6/
     X       5X'TH2  ', f15.6, 7x, f10.6/
     X       5X'PS2  ', f15.6, 7x, f10.6/
     X       5X'PHC  ', f15.6, 7x, f10.6/
     X       5X'THC  ', f15.6, 7x, f10.6/
     X       5X'LC   ', f15.6, 7x, f10.6// )

 301  FORMAT(5X'R1SA1                 '3f8.2)
 302  FORMAT(5X'KS, CS, NC, DETG    ' 2F8.1,I4,F8.4/
     X       5X'TAUJ, LCJ, LCJO     ' 8(F8.1,3X,2F10.4/25X))
 303  FORMAT(5X'KSWRL,KDYN,KR2,KPS2 ' 4F8.2)
 305  FORMAT('TRIM'/
     X       5X'R1SN              ',  3F9.1/
     X       5X'V1SN               ', 3F9.2/
     X       5X'PH1, TH1, PS1 (deg)', 3F9.2/
     X       5X'PH2, TH2, PS2 (deg)', 3F9.2/
     X       5X'OM22 (rps)         ', 3f9.3/
     X       5X'DV2SN (fps2)       ', 3f9.3/
     X       5X'RA2S2              ', 3F9.2/
     X       5X'VA2SN              ', 3F9.2/
     X       5X'FA11, FA1N        ',  6F9.1/
     X       5X'MA11              ',  3F9.1/
     X       5X'FC11, FC1N        ',  6F9.1/
     X       5X'MC11              ',  3F9.1/
     X       5X'FA22, FA2N        ',  6F9.1/
     X       5X'MA22              ',  3F9.1/
     X       5X'ALF2D, BET2D (deg) ', 2F9.2/
     X       5X'TRAIL ANGLE (deg) ',  F10.2/
     X       5X'da, db, dc, dp   ',   4F9.3)

 580  FORMAT(/
     X'   RUN INFORMATION AND TRIM DATA WRITTEN TO ',A/
     X'   ',1F7.2,' SEC SIMULATION RECORD WRITTEN TO ',A/)

 590  TYPE *,'error in reading data array'
      WRITE(6,591) I
 591  FORMAT('stopped at record number',I8)
      CALL closeW(unit)
      GO TO 599
 592  TYPE *,'error in reading data array'
      WRITE(6,593) I
 593  FORMAT('stopped at record number',I8)
      CALL closeW(unit)

 680  FORMAT(/
     X'   WAKE INFORMATION WRITTEN TO ',A/)
```

END

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX O SLVARS.CMN

Common block variables for GenHel / Slung Load simulation. Stabilization parameters were added. Program by Tyson, P. modified by Ehlers, G.

```
C file genhel/batch/sl/slvars.cmn  ------------------------- 31 AUG 98
Peter Tyson
C    Load stabil parameters added 17 Jul 01 by George Ehlers

C  Common blocks for the GenHel/Slung Load Simulation.  variables
required for data output from the No-Load,
C  SLSC, and SLMC cases.   The common blocks occurring herein are:

C   /LFLOAT/RLOAD(500)        load/sling parameters and variables
C   /LFIXED/LLOAD(15)         discretes for slung load codes
C   /XFLOAT/A(500)            Strike variables
C   /IFLOAT/IA(500)           Stike descretes
C   /FCSCOM/FCS(100)          FCS
C   /ROCOM/RC(440)            Wake varbs/params
C   /IRCOM/IRC(20)              ?
C   /RCOM/RO(70)        pilot inputs
C   /CSLLNCMN/LOADNAME
C   /CSLCFCMN/CHFILE
C   /CSLDFCMN/DATAFILE

      REAL
     X   PH2DEG, TH2DEG, PS2DEG, ALF2D, BET2D, OM2N(3), V2SN(3),
     X   VA2SN(3), VA2S2(3), FA11(3), FA1N(3), MA11(3), FC11(3),
     X   FC1N(3), MC11(3), FA22(3), FA2N(3), MA22(3), FC22(3),
     X   FC2N(3), MC22(3), FA2W(3), MA2W(3), W2, I2XX, I2YY, I2ZZ,
     X   I2XZ, KS, CS, DOQ, LCJ(8), LCJO(8), TAUJ(8), R2SJ2(3,8),
     X   RA2P2(3), R2P2S2(3), RA2PO2(3), R2P2SO2(3), R2PJ2(3,8),
     X   TAU, RSN(3), DELRSN(3), R2SN(3), VSN(3), DELVSN(3),
     X   DV2SN(3), DELDVSN(3), SSFN(3), ISSFN(3), IISSFN(3),
     X   OMCC(3), LCO, LCMLCO, PHCH, THCH, RPHC, RTHC, R2S12(3),
     X   RA2S1(3), TOW, FWT, GAM, PSVA,
     X   DETG, R1SA1(3), DELPS2O, ALFO, PSWO, FREQ,
     X   QO(12), DU(12), U(12), DQ(12), Q(12), DV1SN(3), DOM11(3),
     X   DOM22(3), DDRA2S2(3), OM11(3), OM22(3), DRA2S2(3), DPH1,
     X   DTH1, DPS1, DPH2, DTH2, DPS2, R1SN(3), PH1, TH1, PH2, TH2,
     X   PS2, RA2S2(3), PH1DEG, TH1DEG, PS1DEG, WN(3), G, I1XX,
     X   I1YY, I1ZZ, I1XZ, DT, W1, RHO, T, R2D, XMUXH, XMUYH, XMUZH,
     X   XMUXS, XMUYS, XMUZS, XLAMDA, CTA, OMEGA, A1F, B1F, FSCG,
     X   WLCG, BLCG, RMR, T2N(3,3), TN2(3,3), DALF2D, DBET2D, DANG2D,
     X   ALFAO, BETAO, R1S2SN(3), P2N, Q2N, R2N, P21, Q21, R21,
     X   CAD_SAVE(4),R2SS2(3),AMGS2(3),DV1S1(3),
     X   R2OD,KSWRL,KR2,KPS2,KDYN,PWL_TIME(20),PWLCNTRL(20),
     X   TEMP, GAINH, CH, SH, CLAH, CLDH, CDOH, CDPH, STALH1,
     X   STALH2, RH(3), GAINV, CV, SV, CLAV, CLDV, CDOV, CDPV,
     X   STALV1, STALV2, RV(3), F1f, F2f, F3f, m1f, m2f, m3f
```

```
      INTEGER STRETCH, NSTORE, NREC, NC, QUAD, AXIS,STABSYST,ACTIVE

      COMMON /LFLOAT/ RLOAD(500)

C LOAD EULER ROLL,PITCH AND YAW ANGLES - DEG
      EQUIVALENCE (RLOAD(  1), PH2DEG )
      EQUIVALENCE (RLOAD(  2), TH2DEG )
      EQUIVALENCE (RLOAD(  3), PS2DEG )

C LOAD ANGLES OF ATTACK AND SIDESLIP - DEG
      EQUIVALENCE (RLOAD(  4), ALF2D )
      EQUIVALENCE (RLOAD(  5), BET2D )

C LOAD ROLL, PITCH AND YAW RATES, INERTIAL FRAME - RAD/SEC
      EQUIVALENCE (RLOAD(  6), OM2N(1) )

C LOAD ROLL, PITCH AND YAW RATES, LOAD-HC HEADING COORDS- RAD/SEC
      EQUIVALENCE (RLOAD(  9), P2P )
      EQUIVALENCE (RLOAD( 10), Q2P )

C LOAD VELOCITIES, INERTIAL FRAME - FT/SEC
      EQUIVALENCE (RLOAD( 11), V2SN(1) )

C LOAD APPARENT WIND VELOCITIES, INERTIAL FRAME - FT/SEC
      EQUIVALENCE (RLOAD( 14), VA2SN(1) )

C LOAD APPARENT WIND VELOCITIES, LOAD BODY FRAME - FT/SEC
      EQUIVALENCE (RLOAD( 17), VA2S2(1) )

C HELO AERO FORCES, HC BODY AXES
      EQUIVALENCE (RLOAD( 20), FA11(1) )

C HELO AERO FORCES, INERTIAL AXES
      EQUIVALENCE (RLOAD( 23), FA1N(1) )

C HELO AERO MOMENTS ABOUT HC CG, HC BODY AXES
      EQUIVALENCE (RLOAD( 26), MA11(1) )

C HOOK FORCES, HC BODY AXES
      EQUIVALENCE (RLOAD( 29), FC11(1) )

C HOOK FORCES, INERTIAL AXES
      EQUIVALENCE (RLOAD( 32), FC1N(1) )

C HOOK MOMENTS ABOUT HC CG, HC BODY AXES
      EQUIVALENCE (RLOAD( 35), MC11(1) )

C LOAD AERO FORCES, LOAD BODY AXES
      EQUIVALENCE (RLOAD( 38), FA22(1) )

C LOAD AERO FORCES, INERTIAL AXES
      EQUIVALENCE (RLOAD( 41), FA2N(1) )

C LOAD AERO MOMENTS ABOUT LOAD CG, LOAD BODY AXES
      EQUIVALENCE (RLOAD( 44), MA22(1) )

C SLING FORCES, LOAD BODY AXES
```

```
      EQUIVALENCE (RLOAD( 47), FC22(1) )

C SLING FORCE, INERTIAL AXES
      EQUIVALENCE (RLOAD( 50), FC2N(1) )

C SLING MOMENTS ABOUT LOAD CG, LOAD BODY AXES
      EQUIVALENCE (RLOAD( 53), MC22(1) )

C LOAD AERO FORCES, LOAD APPARENT WIND AXES
      EQUIVALENCE (RLOAD( 56), FA2W(1) )

C LOAD AERO MOMENTS ABOUT LOAD CG, LOAD APPARENT WIND AXES
      EQUIVALENCE (RLOAD( 59), MA2W(1) )

C LOAD INITIAL YAW RATE
      EQUIVALENCE (RLOAD( 62), R2OD )

C LOAD DYNAMIC AERODYNAMICS CONSTANTS
      EQUIVALENCE (RLOAD( 63), KSWRL )
      EQUIVALENCE (RLOAD( 64), KR2 )
      EQUIVALENCE (RLOAD( 65), KPS2 )
      EQUIVALENCE (RLOAD( 66), KDYN )

C RLOAD( 67) - ( 69) EMPTY

C LOAD WEIGHT (POUNDS)
      EQUIVALENCE (RLOAD( 70), W2 )

C LOAD XX, YY, ZZ, AND XZ MOMENTS OF INERTIA - SLUG-FT2
      EQUIVALENCE (RLOAD( 71), I2XX )
      EQUIVALENCE (RLOAD( 72), I2YY )
      EQUIVALENCE (RLOAD( 73), I2ZZ )
      EQUIVALENCE (RLOAD( 74), I2XZ )

C SLING SPRING CONSTANT
      EQUIVALENCE (RLOAD( 75), KS )

C SLING DAMPING CONSTANT
      EQUIVALENCE (RLOAD( 76), CS )

C LOAD DRAG OVER DYNAMIC PRESSURE
      EQUIVALENCE (RLOAD( 77), DOQ )

C SLING CABLE LENGTH (1 TO 8 LEGS)
      EQUIVALENCE (RLOAD( 78), LCJ(1) )

C SLING IC LENGTH (1 TO 8 LEGS)
      EQUIVALENCE (RLOAD( 86), LCJO(1) )

C SLING CABLE TENSIONS
      EQUIVALENCE (RLOAD( 94), TAUJ(1) )

      EQUIVALENCE (RLOAD(103), R2SJ2(1,1) )
      EQUIVALENCE (RLOAD(127), RA2P2(1) )
      EQUIVALENCE (RLOAD(130), R2P2S2(1) )
      EQUIVALENCE (RLOAD(133), RA2PO2(1) )
      EQUIVALENCE (RLOAD(136), R2P2SO2(1) )
```

```
      EQUIVALENCE (RLOAD(139), R2PJ2(1,1) )

C SLSC VARIABLES

C SLING TENSION
      EQUIVALENCE (RLOAD(164), TAU )

C SYSTEM MOTION WRT INERTIAL FRAME, FT
      EQUIVALENCE (RLOAD(165), RSN(1) )

C CHANGE IN SYSTEM MOTION WRT INERTIAL FRAME, FT
      EQUIVALENCE (RLOAD(168), DELRSN(1) )

C LOAD MOTION WRT INERTIAL FRAME, FT
      EQUIVALENCE (RLOAD(171), R2SN(1) )

C SYSTEM VELOCITY WRT INERTIAL FRAME, FT/S
      EQUIVALENCE (RLOAD(174), VSN(1) )

C CHANGE IN SYSTEM VELOCITY WRT INERTIAL FRAME, FT/S
      EQUIVALENCE (RLOAD(177), DELVSN(1) )

C LOAD ACCELERATION WRT INERTIAL FRAME, FT/S**2
      EQUIVALENCE (RLOAD(180), DV2SN(1) )

C CHANGE IN SYSTEM ACCELERATION WRT INERTIAL FRAME, FT/S**2
      EQUIVALENCE (RLOAD(183), DELDVSN(1) )

      EQUIVALENCE (RLOAD(186), SSFN(1) )
      EQUIVALENCE (RLOAD(189), ISSFN(1) )
      EQUIVALENCE (RLOAD(192), IISSFN(1) )
      EQUIVALENCE (RLOAD(195), OMCC(1) )
      EQUIVALENCE (RLOAD(198), LCO )
      EQUIVALENCE (RLOAD(200), LCMLCO )
      EQUIVALENCE (RLOAD(201), PHCH )
      EQUIVALENCE (RLOAD(202), THCH )
      EQUIVALENCE (RLOAD(203), RPHC )
      EQUIVALENCE (RLOAD(204), RTHC )

C LOCATION VECTOR, FROM HC CG TO LOAD CG, INERTIAL AXES
      EQUIVALENCE (RLOAD(205), R1S2SN(1) )

C LOCATION VECTOR
      EQUIVALENCE (RLOAD(240), R2S12(1) )

C LOCATION VECTOR, LOAD C.G. FROM HC HOOK, HC BODY AXES
      EQUIVALENCE (RLOAD(243), RA2S1(1) )

C HELICOPTER TAKEOFF WEIGHT
      EQUIVALENCE (RLOAD(246), TOW )

C HELICOPTER TAKEOFF FUEL WEIGHT
      EQUIVALENCE (RLOAD(247), FWT )

      EQUIVALENCE (RLOAD(248), GAM )
      EQUIVALENCE (RLOAD(249), PSVA )
      EQUIVALENCE (RLOAD(250), DETG )
```

```
      EQUIVALENCE (RLOAD(251), R1SA1(1) )
      EQUIVALENCE (RLOAD(254), DELPS2O )
      EQUIVALENCE (RLOAD(255), ALFO )
      EQUIVALENCE (RLOAD(256), PSWO )
      EQUIVALENCE (RLOAD(257), FREQ )

C DYNAMIC AERODYNAMICS PROPORTIONALITY FACTOR
      EQUIVALENCE (RLOAD(258), DYNAMIC )

C ANGLE OF ATTACK, SIDESLIP AND TOTAL ANGULAR RATE OF CHANGE
      EQUIVALENCE (RLOAD(259), DALF2D )
      EQUIVALENCE (RLOAD(260), DBET2D )
      EQUIVALENCE (RLOAD(261), DANG2D )

C FORCE ON LOAD DUE TO ANGULAR RATES OF CHANGE
      EQUIVALENCE (RLOAD(262), FDALF2D )
      EQUIVALENCE (RLOAD(263), FDBET2D )
      EQUIVALENCE (RLOAD(264), FDANG2D )

C QUADRANT 1 ANGLE OF ATTACK AND SIDESLIP FOR LOAD
      EQUIVALENCE (RLOAD(265), ALFAO )
      EQUIVALENCE (RLOAD(266), BETAO )

C LOAD MOTION, INERTIAL AXES
      EQUIVALENCE (RLOAD(267), P2N )
      EQUIVALENCE (RLOAD(268), Q2N )
      EQUIVALENCE (RLOAD(269), R2N )

C LOAD MOTION, HC BODY AXES
      EQUIVALENCE (RLOAD(270), P21 )
      EQUIVALENCE (RLOAD(271), Q21 )
      EQUIVALENCE (RLOAD(272), R21 )

C stabilizing control (feedfwd (cad-save) and feedback) frm pilot.f
C load cg-to-accelerometer location and load accelerometer signals
      EQUIVALENCE (RLOAD(273), DELXAAD)
      EQUIVALENCE (RLOAD(274), DELXBAD)
      EQUIVALENCE (RLOAD(275), DELXPAD)
      EQUIVALENCE (RLOAD(276), DELXCAD)
      EQUIVALENCE (RLOAD(277), CAD_SAVE(1))
      EQUIVALENCE (RLOAD(281), R2SS2(1))
      EQUIVALENCE (RLOAD(284), AMGS2(1))

C rload(287-288) empty

      EQUIVALENCE (RLOAD(289), QO(1) )

C     state array equivalences for integration
C     du = {dv1sn(3),   dom11(3),    dom22(3),  ddra2s2(3)}
C     u  = { v1sn(3),   om11(3),     om22(3),    dra2s2(3)}
C     q  = { r1sn(3), ph1,th1,ps1, ph2,th2,ps2,   ra2s2(3)}

      EQUIVALENCE (RLOAD(301), DU(12) )
      EQUIVALENCE (RLOAD(321), U(12) )
      EQUIVALENCE (RLOAD(341), DQ(12) )
      EQUIVALENCE (RLOAD(361), Q(12) )
      EQUIVALENCE (RLOAD(362), DV1S1(1) )
```

```
C piece-wise linear control input arrays: 370-399
      EQUIVALENCE (RLOAD(370), PWL_TIME(1))
      EQUIVALENCE (RLOAD(390), PWLCNTRL(1))

C Load Stabilization Parameters
      EQUIVALENCE (RLOAD(460), F1V)
      EQUIVALENCE (RLOAD(461), F2V)
      EQUIVALENCE (RLOAD(462), F3V)
      EQUIVALENCE (RLOAD(463), F1H)
      EQUIVALENCE (RLOAD(464), F2H)
      EQUIVALENCE (RLOAD(465), F3H)

      EQUIVALENCE (RLOAD(470), ACTIVE)
      EQUIVALENCE (RLOAD(471), TEMP)

      EQUIVALENCE (RLOAD(475), GAINH)
      EQUIVALENCE (RLOAD(476), CH)
      EQUIVALENCE (RLOAD(477), SH)
      EQUIVALENCE (RLOAD(478), CLAH)
      EQUIVALENCE (RLOAD(479), CLDH)
      EQUIVALENCE (RLOAD(480), CDOH)
      EQUIVALENCE (RLOAD(481), CDPH)
      EQUIVALENCE (RLOAD(482), STALH1)
      EQUIVALENCE (RLOAD(483), STALH2)
      EQUIVALENCE (RLOAD(484), RH(1))
      EQUIVALENCE (RLOAD(485), RH(2))
      EQUIVALENCE (RLOAD(486), RH(3))

      EQUIVALENCE (RLOAD(487), GAINV)
      EQUIVALENCE (RLOAD(488), CV)
      EQUIVALENCE (RLOAD(489), SV)
      EQUIVALENCE (RLOAD(490), CLAV)
      EQUIVALENCE (RLOAD(491), CLDV)
      EQUIVALENCE (RLOAD(492), CDOV)
      EQUIVALENCE (RLOAD(493), CDPV)
      EQUIVALENCE (RLOAD(494), STALV1)
      EQUIVALENCE (RLOAD(495), STALV2)
      EQUIVALENCE (RLOAD(496), RV(1))
      EQUIVALENCE (RLOAD(497), RV(2))
      EQUIVALENCE (RLOAD(498), RV(3))

C HC ACCELERATIONS, INERTIAL AXES - FT/S2
      EQUIVALENCE (DU( 1),  DV1SN(1) )

C HC BODY ROLL, PITCH AND YAW ACCELERATIONS - RAD/SEC2
      EQUIVALENCE (DU( 4),  DOM11(1) )

C LOAD BODY ROLL, PITCH AND YAW ACCELERATIONS - RAD/SEC2
      EQUIVALENCE (DU( 7),  DOM22(1) )

C STRETCH COORDS, HOOK TO LOAD CG LINE SEGMENT COORDS
      EQUIVALENCE (DU(10),  DDRA2S2(1) )


C HC BODY ROLL, PITCH AND YAW RATES - RAD/SEC
      EQUIVALENCE (U( 4),   OM11(1) )
```

```
C LOAD BODY ROLL, PITCH AND YAW RATES - RAD/SEC
      EQUIVALENCE (U( 7),   OM22(1) )

C FIRST DERIVATIVE, HOOK TO LOAD CG LINE SEGMENT
      EQUIVALENCE (U(10),   DRA2S2(1) )


C HC EULER ROLL, PITCH AND YAW RATES - RAD/SEC
      EQUIVALENCE (DQ( 4),  DPH1 )
      EQUIVALENCE (DQ( 5),  DTH1 )
      EQUIVALENCE (DQ( 6),  DPS1 )

C LOAD EULER ROLL, PITCH AND YAW RATES - RAD/SEC
      EQUIVALENCE (DQ( 7),  DPH2 )
      EQUIVALENCE (DQ( 8),  DTH2 )
      EQUIVALENCE (DQ( 9),  DPS2 )


C HC INERTIAL POSITION VECTOR
      EQUIVALENCE (Q( 1),   R1SN(1) )

C HC EULER ROLL, PITCH AND YAW ANGLES - RAD
      EQUIVALENCE (Q( 4),   PH1 )
      EQUIVALENCE (Q( 5),   TH1 )
C Note:  PS1 included separately where needed

C LOAD EULER ROLL, PITCH AND YAW ANGLES - RAD
      EQUIVALENCE (Q( 7),   PH2 )
      EQUIVALENCE (Q( 8),   TH2 )
      EQUIVALENCE (Q( 9),   PS2 )

C HOOK TO LOAD CG LINE SEGMENT, LOAD BODY AXES
      EQUIVALENCE (Q(10),   RA2S2(1) )

C STRIKE Variables
C Used by:  GHSLMC, GHSLMC_IC, CONEXAERO, NRT_UNC3_OUT
      COMMON /XFLOAT/ A(500)
      EQUIVALENCE (A(  1), PH1DEG )
      EQUIVALENCE (A(  2), TH1DEG )
      EQUIVALENCE (A(  3), PS1DEG )
      EQUIVALENCE (A( 76), WN(1) )
      EQUIVALENCE (A( 97), G )
      EQUIVALENCE (A(116), I1XX )
      EQUIVALENCE (A(117), I1YY )
      EQUIVALENCE (A(118), I1ZZ )
      EQUIVALENCE (A(119), I1XZ )
      EQUIVALENCE (A(168), DT )
      EQUIVALENCE (A(183), RHO )
      EQUIVALENCE (A(209), SOUND )
      EQUIVALENCE (A(303), T )
      EQUIVALENCE (A(359), R2D )

C STRIKE Switches
C Used by:  GHSLMC, CONEXAERO
      COMMON /IFIXED/ IA(250)
      EQUIVALENCE (IA(1), IMODE )
```

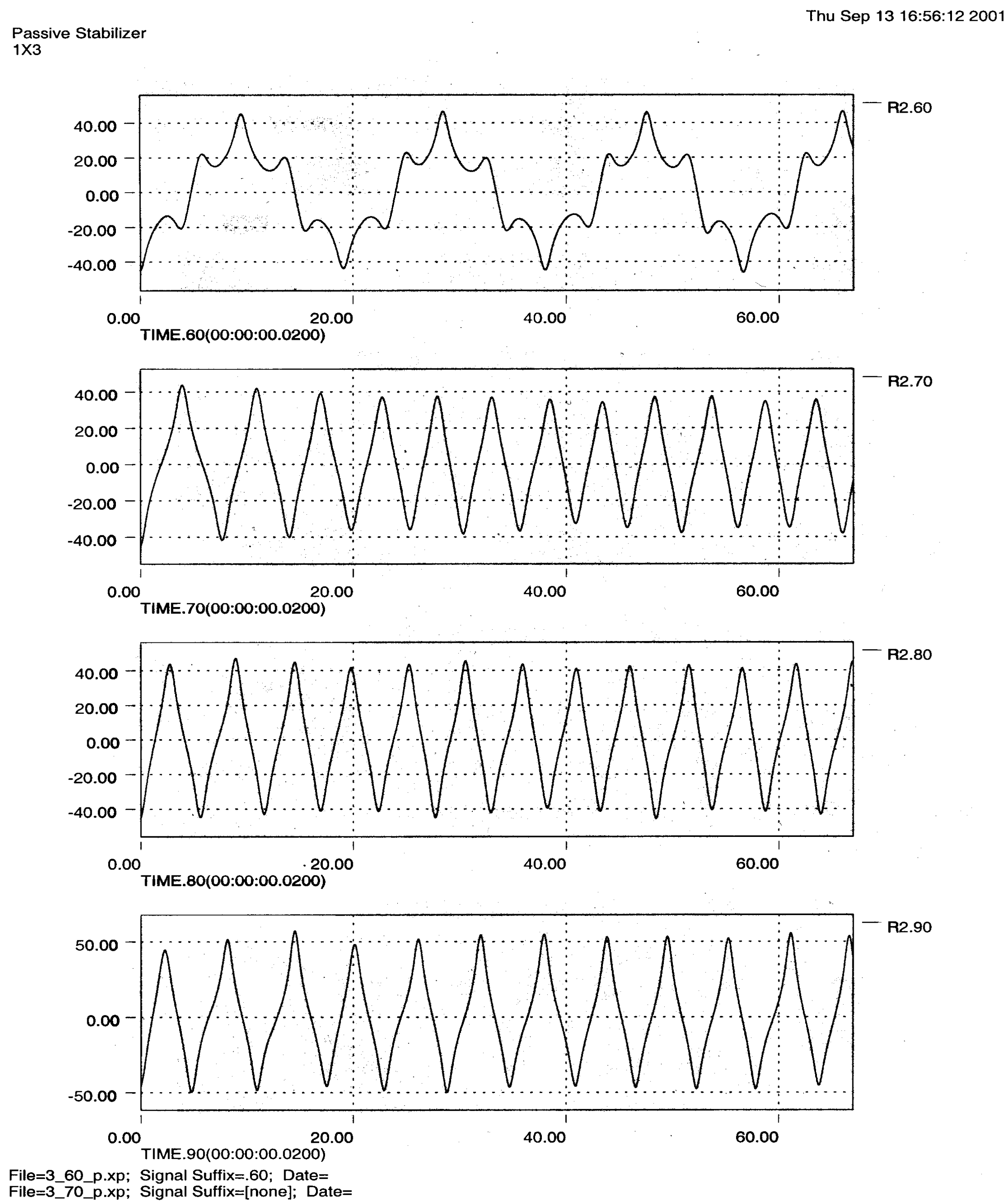
```
C Variables for data output
C Used by:  NRT_UNC3_OUT
      COMMON /FCSCOM/ FCS(100)
      EQUIVALENCE(FCS(16), RSAS)
      EQUIVALENCE(FCS(17), PSAS)
      EQUIVALENCE(FCS(18), YSAS)
      EQUIVALENCE(FCS(41), DMIXA)
      EQUIVALENCE(FCS(42), DMIXB)
      EQUIVALENCE(FCS(43), DMIXC)
      EQUIVALENCE(FCS(44), DMIXP)
      EQUIVALENCE(FCS(49), PSFWD)
      EQUIVALENCE(FCS(50), PSAFT)
      EQUIVALENCE(FCS(51), PSLAT)
      EQUIVALENCE(FCS(52), PSTR)

C Variables for wake computations and locations
C Used by:  CONEXAERO, NRT_UNC3_OUT
      COMMON /ROCOM/  RC(440)
      EQUIVALENCE (RC( 51), XMUXH )
      EQUIVALENCE (RC( 52), XMUYH )
      EQUIVALENCE (RC( 53), XMUZH )
      EQUIVALENCE (RC( 54), XMUXS )
      EQUIVALENCE (RC( 55), XMUYS )
      EQUIVALENCE (RC( 56), XMUZS )
      EQUIVALENCE (RC( 70), XLAMDA )
      EQUIVALENCE (RC( 71), CTA )
      EQUIVALENCE (RC( 80), A1F )
      EQUIVALENCE (RC( 81), B1F )
      EQUIVALENCE (RC(117), OMEGA )
      EQUIVALENCE (RC(218), FSCG )
      EQUIVALENCE (RC(219), WLCG )
      EQUIVALENCE (RC(220), BLCG )
      EQUIVALENCE (RC(221), RMR )
      EQUIVALENCE (RC(223), W1 )
      EQUIVALENCE (RC(280), WBLADE )

      COMMON /IRCOM/ IRC(20)
      EQUIVALENCE (IRC(10), NBS)

C Control History variables
C Used by:  NRT_UNC3_OUT, PILOT
      COMMON /RCON/ RO(70)
      EQUIVALENCE (RO(13), XA )
      EQUIVALENCE (RO(14), XAAD )
      EQUIVALENCE (RO(25), XB )
      EQUIVALENCE (RO(26), XBAD )
      EQUIVALENCE (RO(37), XP )
      EQUIVALENCE (RO(38), XPAD )
      EQUIVALENCE (RO( 1), XC )
      EQUIVALENCE (RO( 2), XCAD )


      COMMON /LFIXED/ LLOAD(15)
      EQUIVALENCE (LLOAD( 1), STRETCH )
      EQUIVALENCE (LLOAD( 2), NSTORE )
      EQUIVALENCE (LLOAD( 3), NC )
```

```
      EQUIVALENCE (LLOAD( 4), IAERSL )
      EQUIVALENCE (LLOAD( 5), ILOAD )
      EQUIVALENCE (LLOAD( 6), IPILOT )
      EQUIVALENCE (LLOAD( 7), IWAKE )
      EQUIVALENCE (LLOAD( 8), ISWIRL )
      EQUIVALENCE (LLOAD( 9), IDATA )
      EQUIVALENCE (LLOAD(10), QUAD )
      EQUIVALENCE (LLOAD(11), AXIS )
      EQUIVALENCE (LLOAD(12), NREC )
      EQUIVALENCE (LLOAD(13), STABSYST)
      EQUIVALENCE (LLOAD(15), ACTIVE)


C Load Description
C Used by:  GHSL_INIT, NRT_UNC3_OUT, NRT_UNC3_IN
      CHARACTER*40 LOADNAME, CHFILE, DATAFILE
      COMMON /CSLLNCMN/ LOADNAME
      COMMON /CSLCFCMN/ CHFILE
      COMMON /CSLDFCMN/ DATAFILE
```

THIS PAGE INTENTIONALLY LEFT BLANK

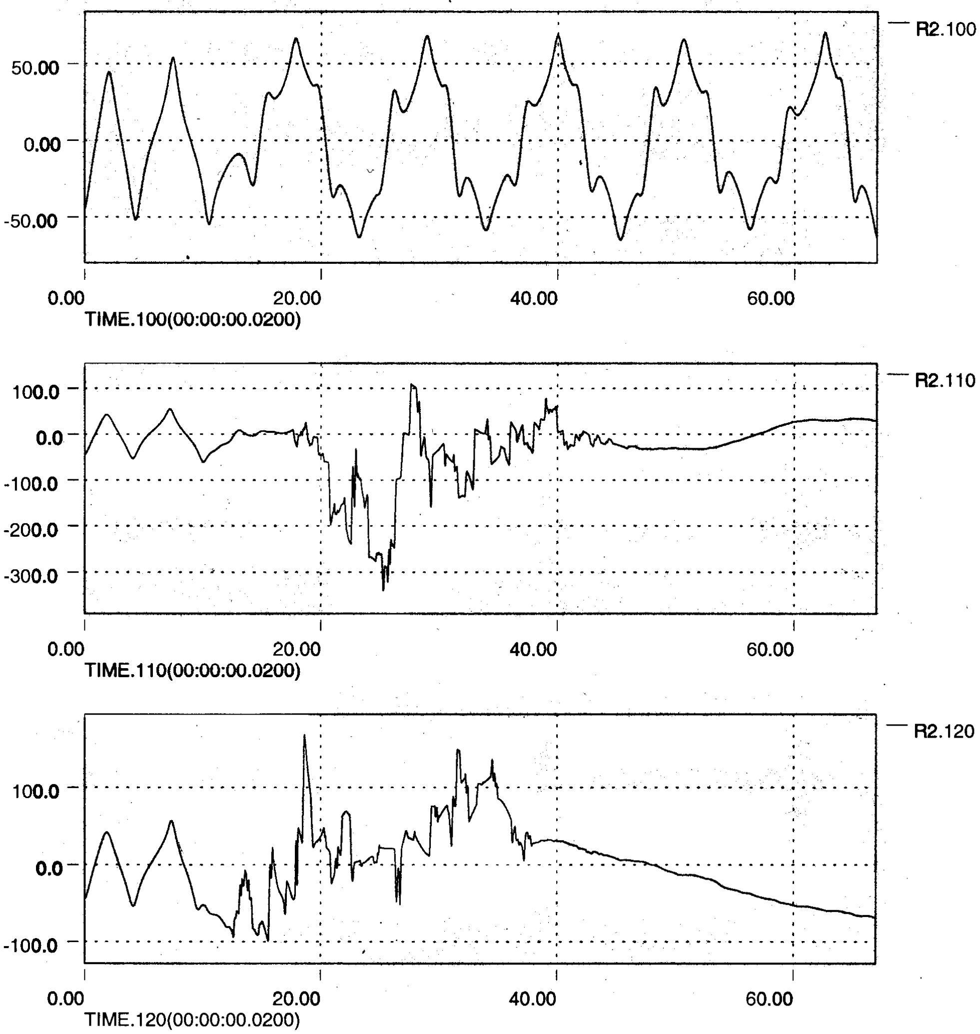
# APPENDIX P STABILIZER RESULTS

Simulation results of passive and active stabilization systems of various areas. Plotted are yaw rate (R2) in feet per second versus time in seconds. Appendix P by Ehlers, G.



Thu Sep 13 16:56:35 2001

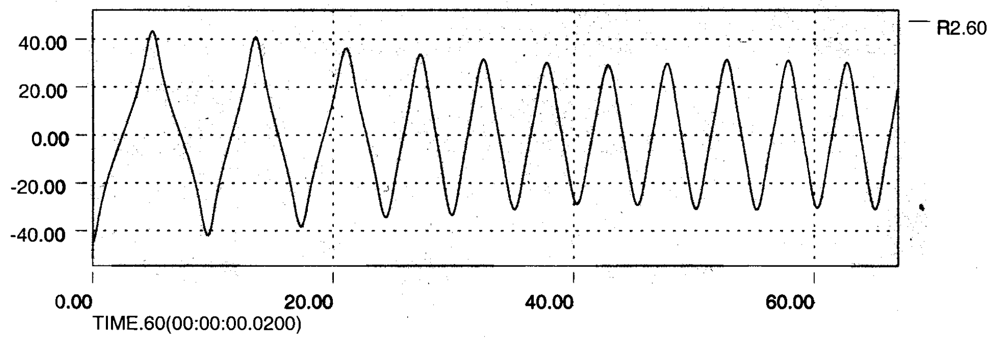
Passive Stabilizer
1X3

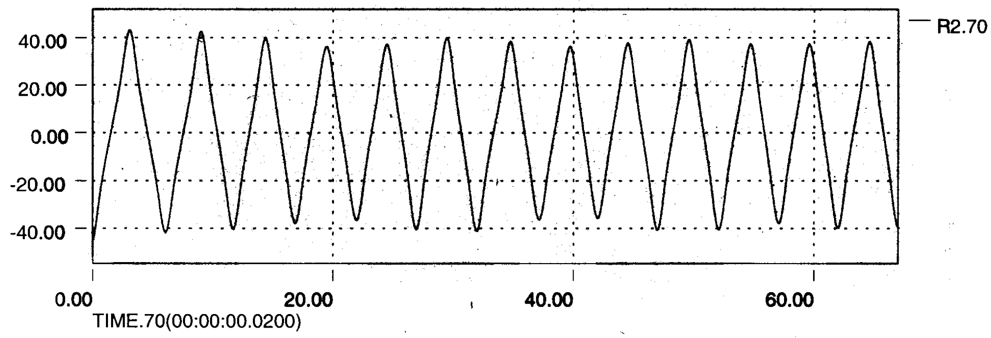


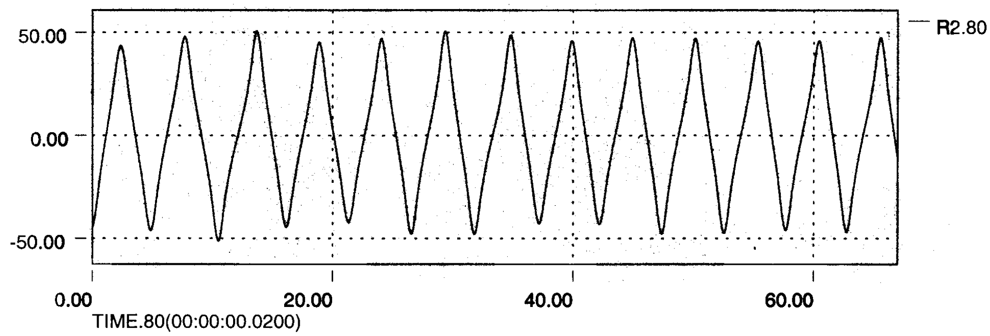

File=3_60_p.xp; Signal Suffix=.60; Date=
File=3_70_p.xp; Signal Suffix=[none]; Date=

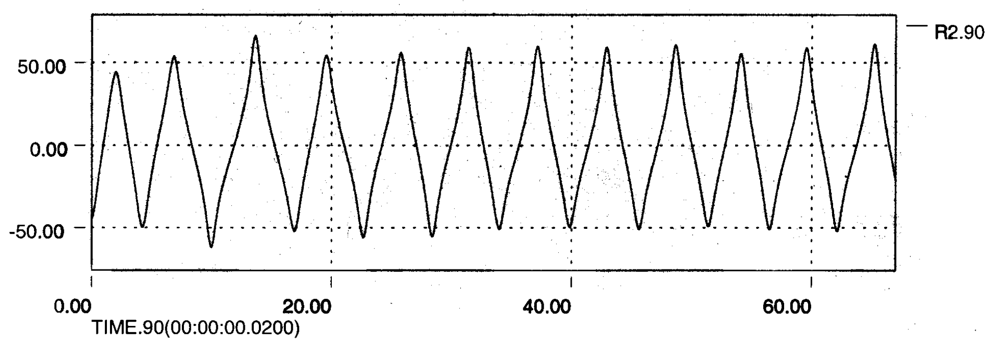
Thu Sep 13 16:57:59 2001

Passive Stabilizer
2X3



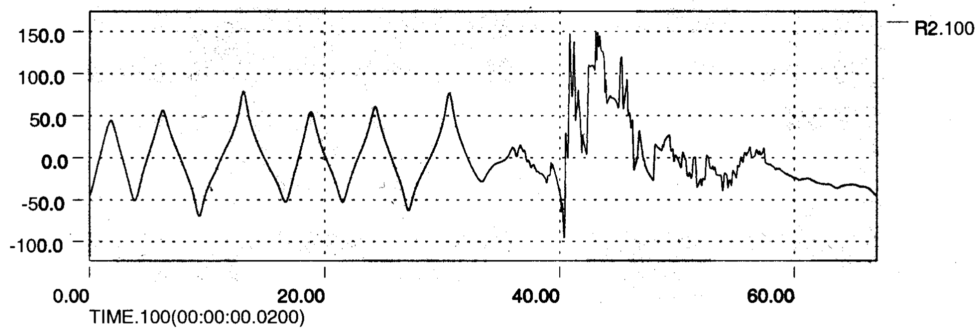



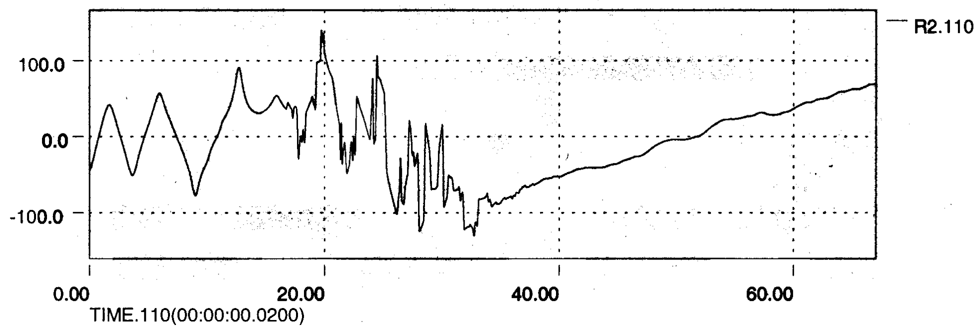



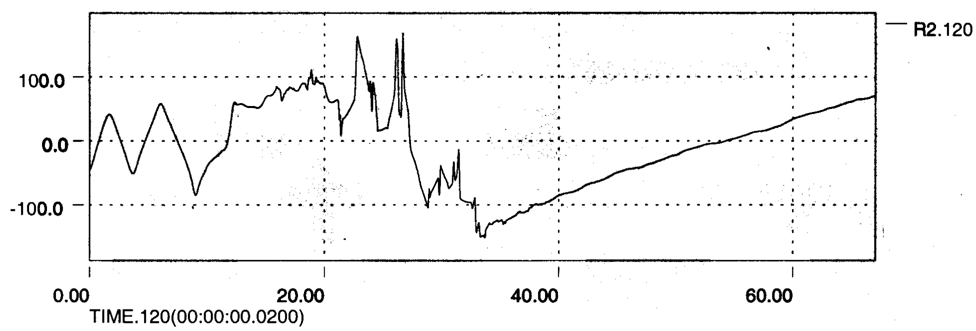

File=6_60_p.xp; Signal Suffix=.60; Date=
File=6_70_p.xp; Signal Suffix=.70; Date=

Thu Sep 13 16:58:22 2001

Passive Stabilizer
2X3



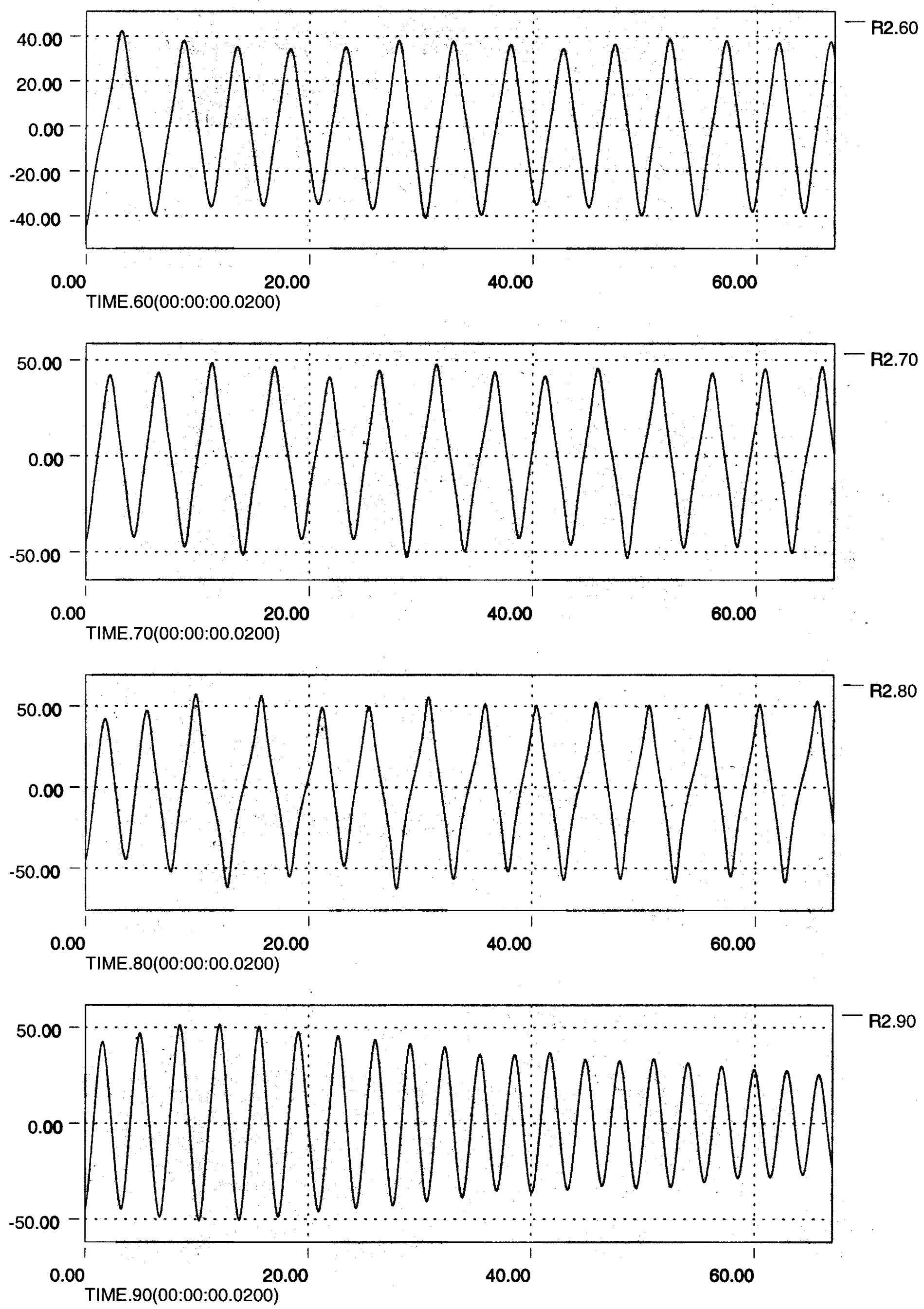




File=6_60_p.xp; Signal Suffix=.60; Date=
File=6_70_p.xp; Signal Suffix=.70; Date=

Thu Sep 13 17:00:48 2001

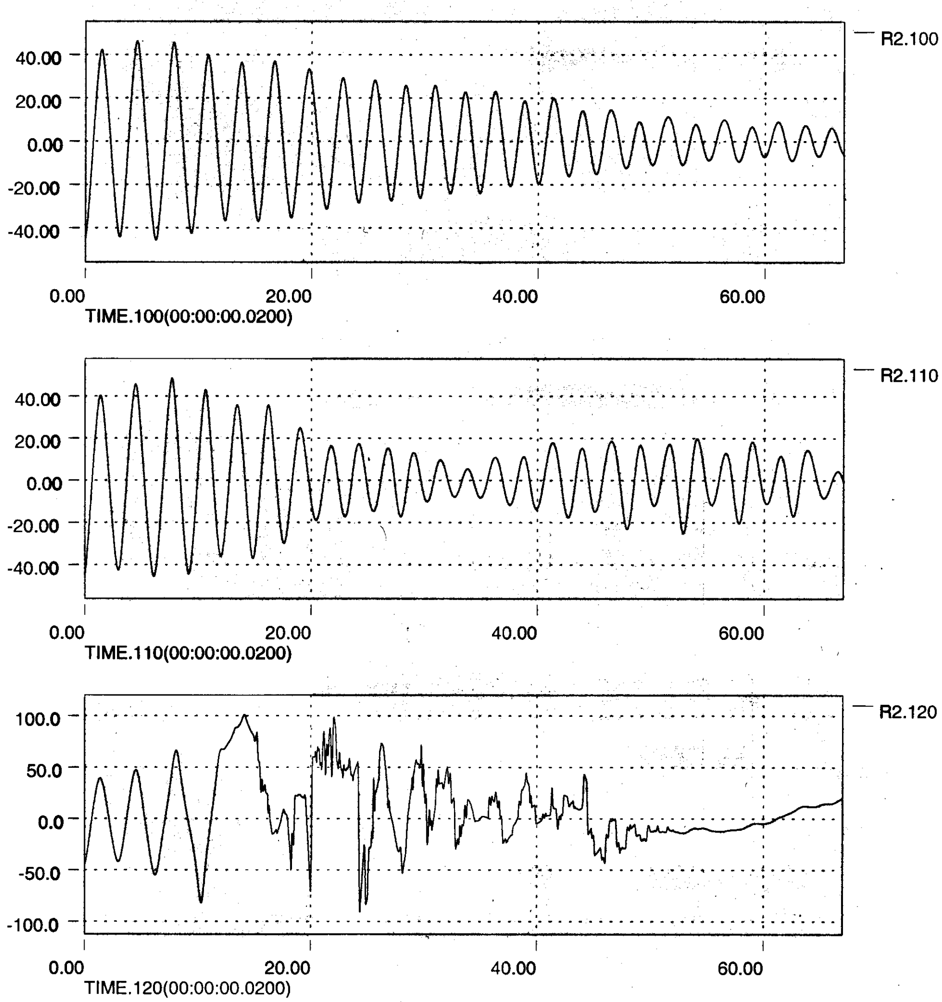
Passive Stabilizer
2X4




File=8_60_p.xp; Signal Suffix=.60; Date=
File=8_70_p.xp; Signal Suffix=.70; Date=

Thu Sep 13 17:01:28 2001

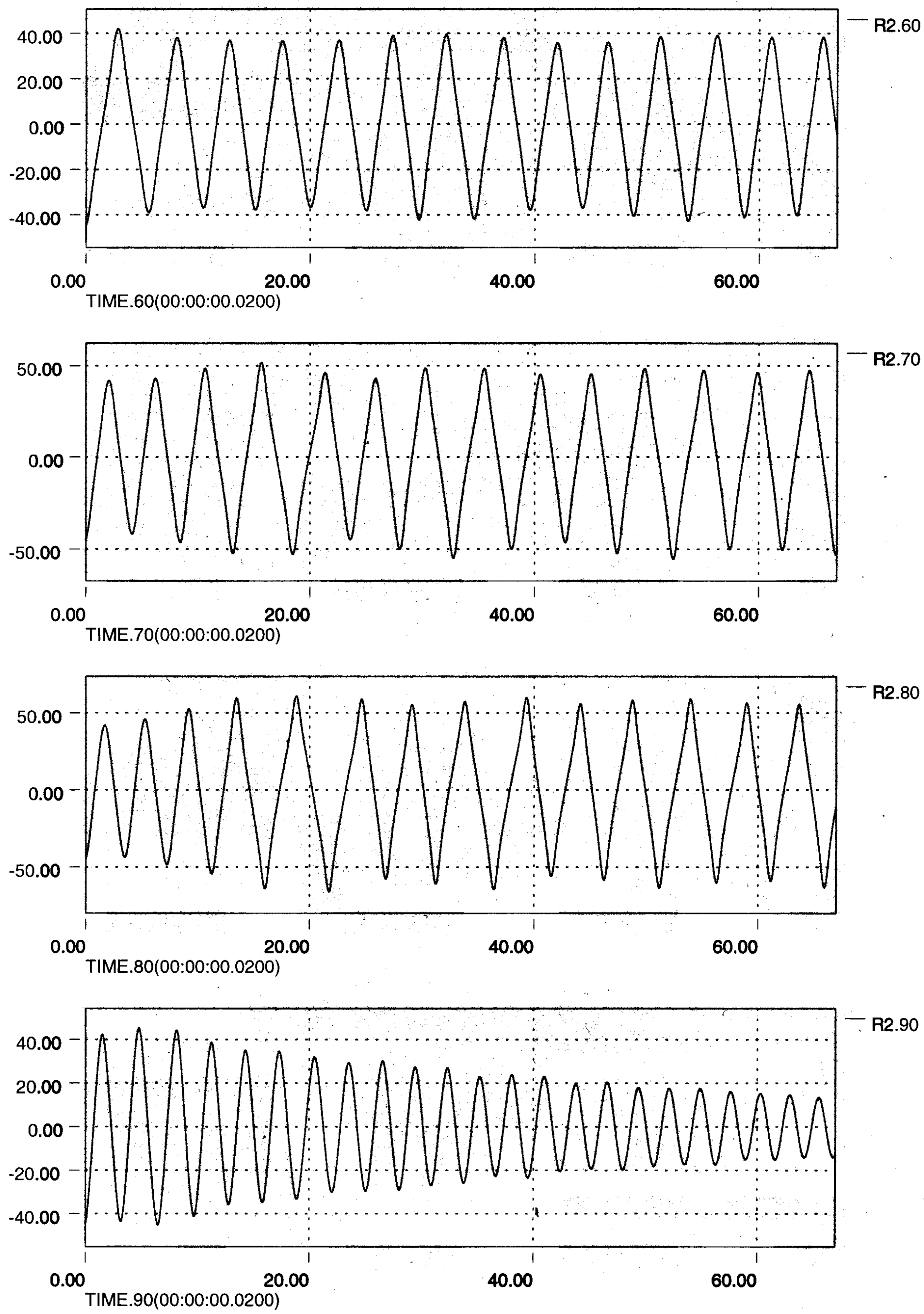
Passive Stabilizer
2X4




File=8_60_p.xp; Signal Suffix=.60; Date=
File=8_70_p.xp; Signal Suffix=.70; Date=

Thu Sep 13 19:14:52 2001

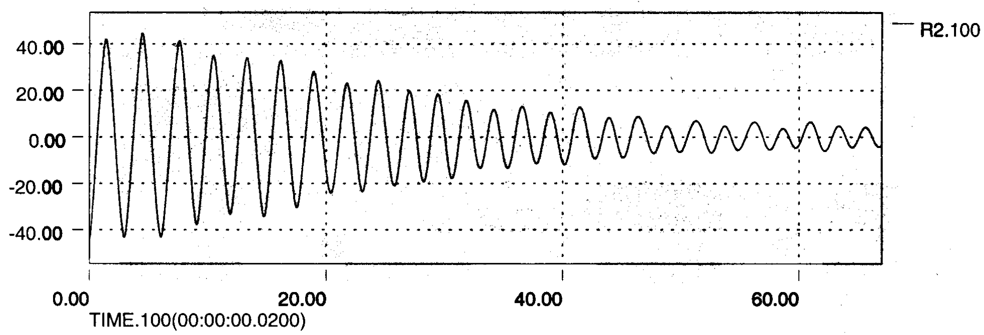
Passive Stabilizer
3X4

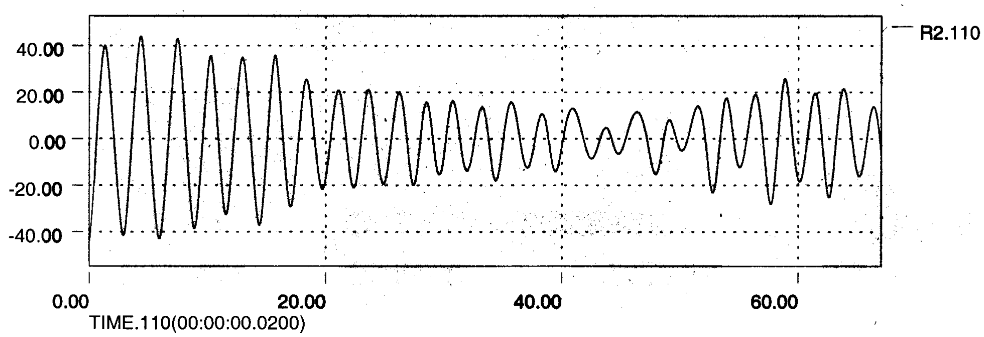


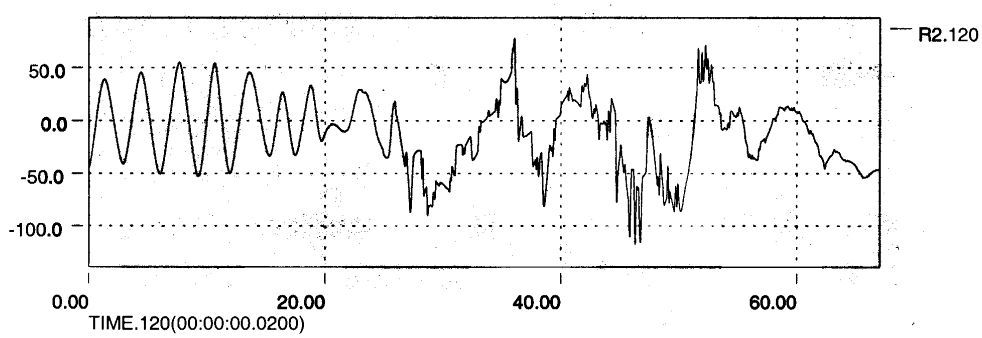

File=12_60_p.xp; Signal Suffix=.60; Date=
File=12_70_p.xp; Signal Suffix=.70; Date=

Thu Sep 13 19:15:13 2001

Passive Stabilizer
3X4



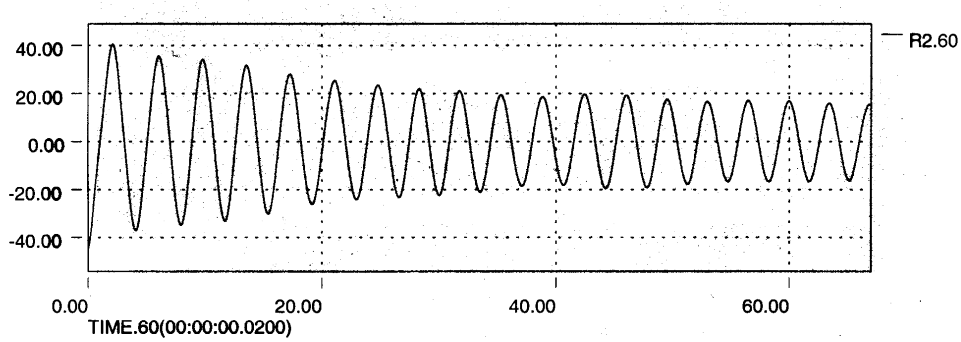



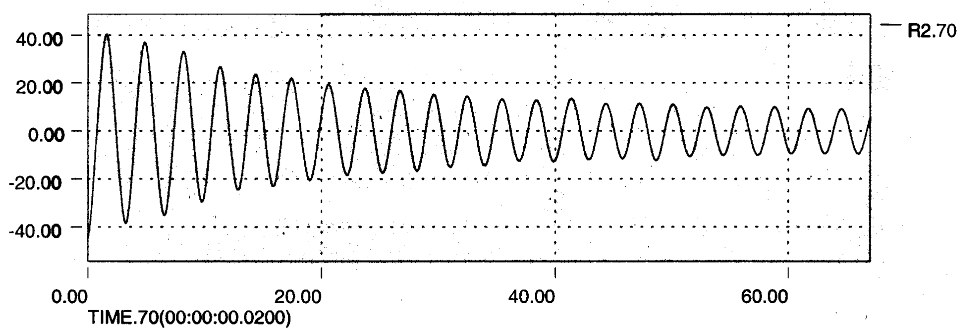



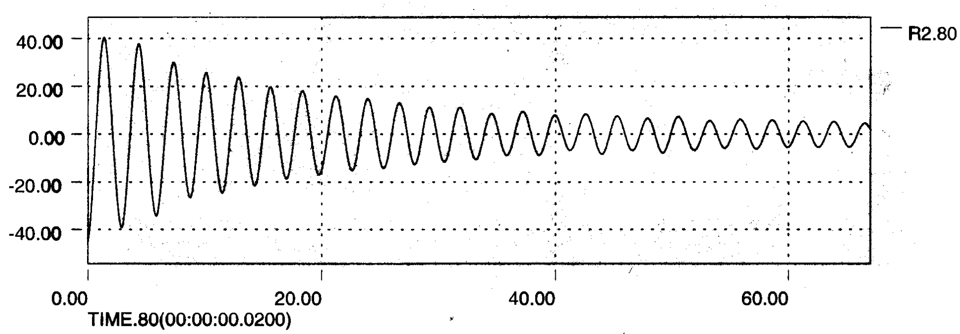

File=12_60_p.xp; Signal Suffix=.60; Date=
File=12_70_p.xp; Signal Suffix=.70; Date=

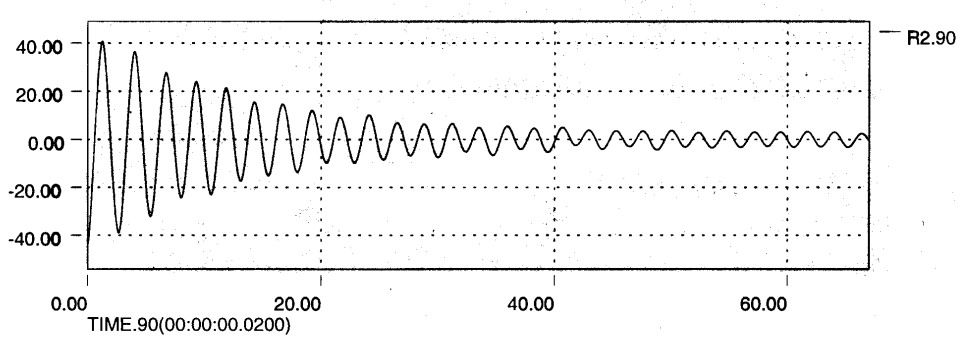
Passive Stabilizer
3X5

Thu Sep 13 16:16:32 2001



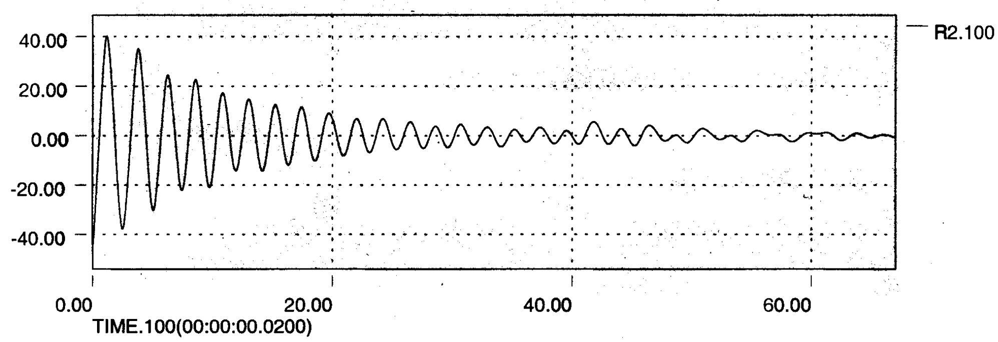



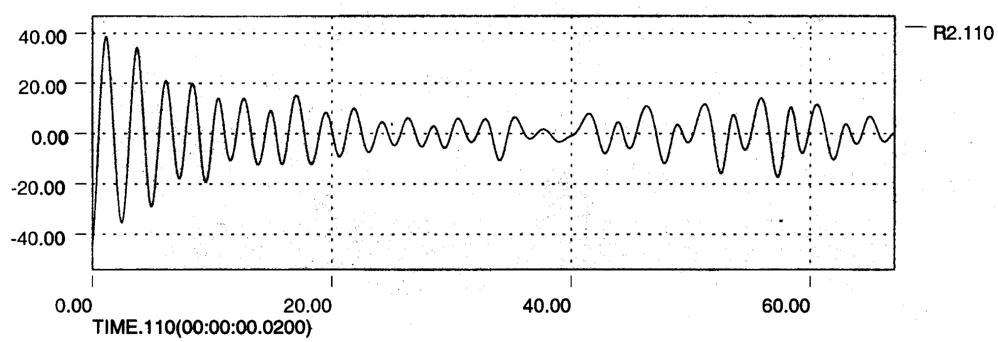



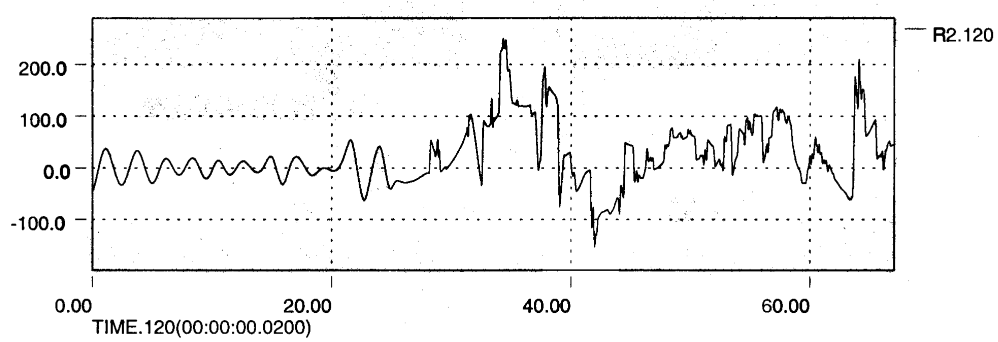

File=15_60_p.xp; Signal Suffix=.60; Date=
File=15_70_p.xp; Signal Suffix=.70; Date=

Thu Sep 13 16:17:28 2001

Passive Stabilizer
3X5



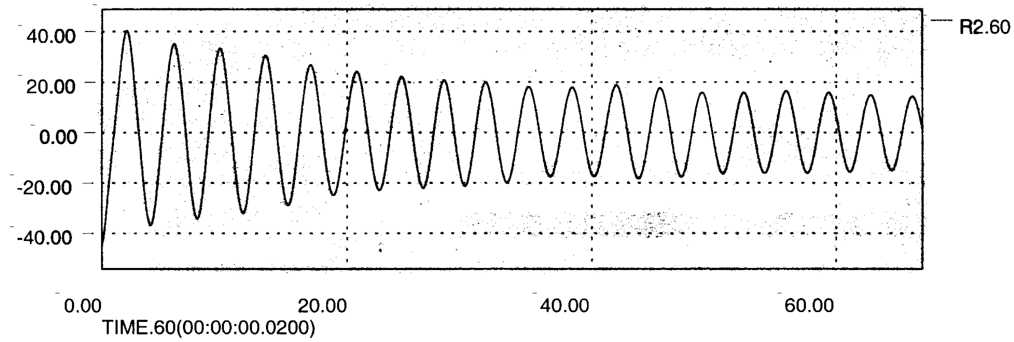



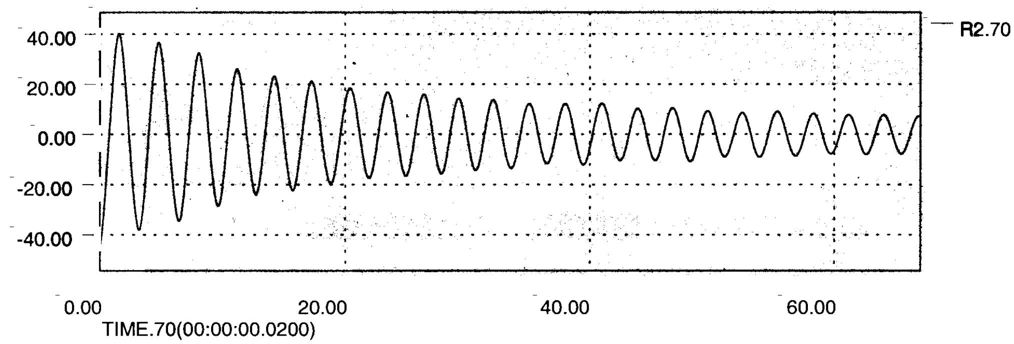



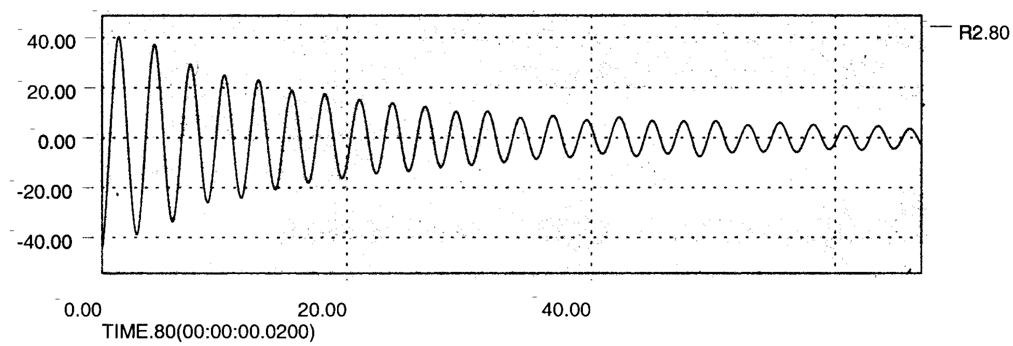

File=15_60_p.xp; Signal Suffix=.60; Date=
File=15_70_p.xp; Signal Suffix=.70; Date=

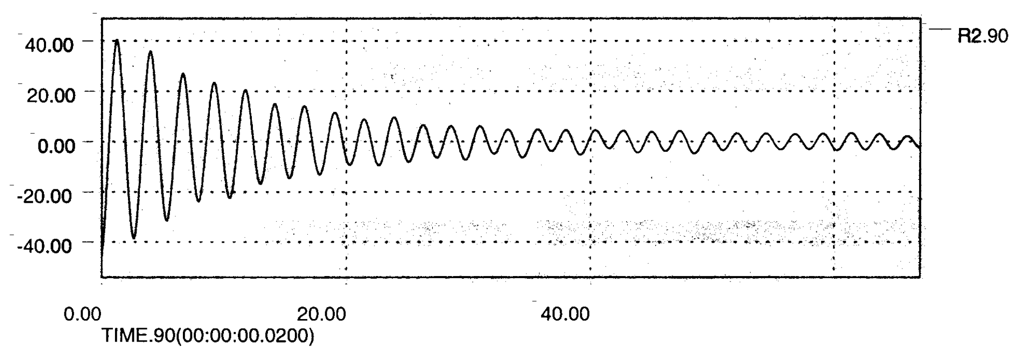
Thu Sep 13 16:37:05 2001

Passive Stabilizer
4X5



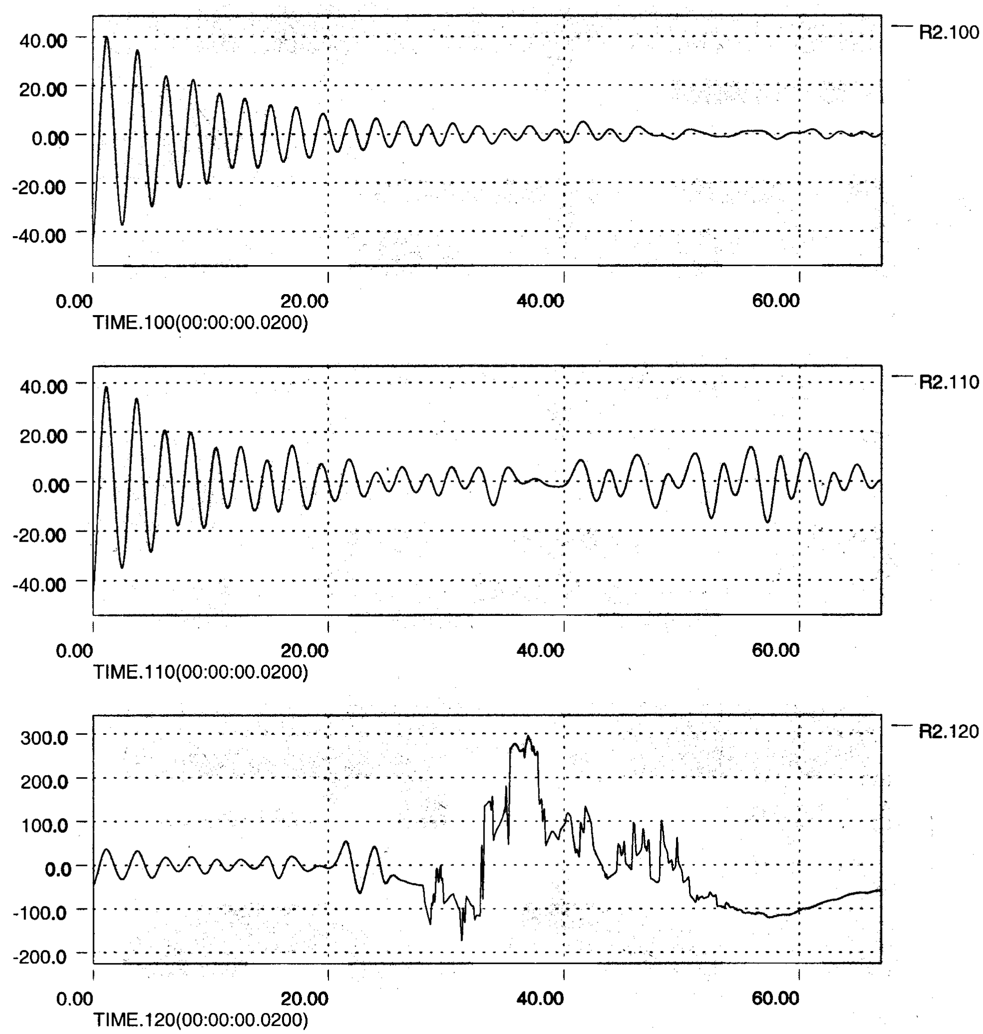




File=20_60_p.xp; Signal Suffix=.60; Date=
File=20_70_p.xp; Signal Suffix=.70; Date=

Thu Sep 13 16:39:37 2001

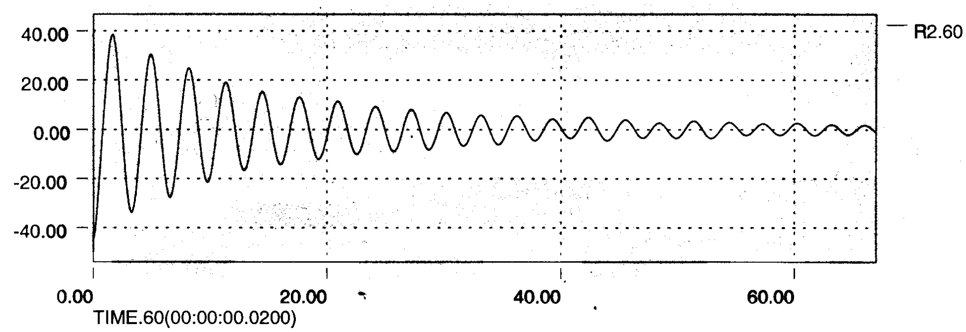
Passive Stabilizer
4X5

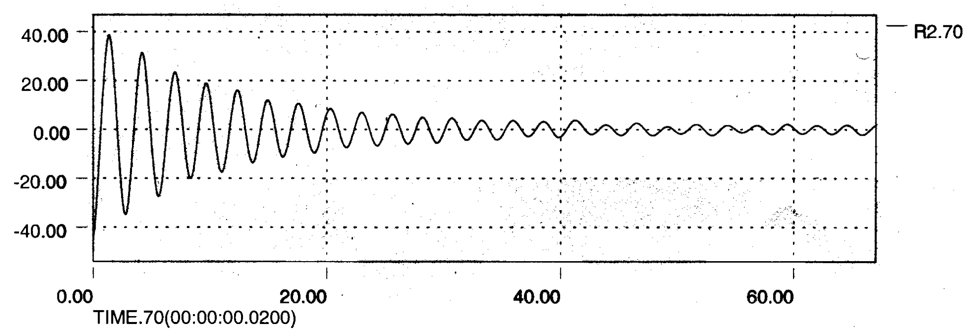


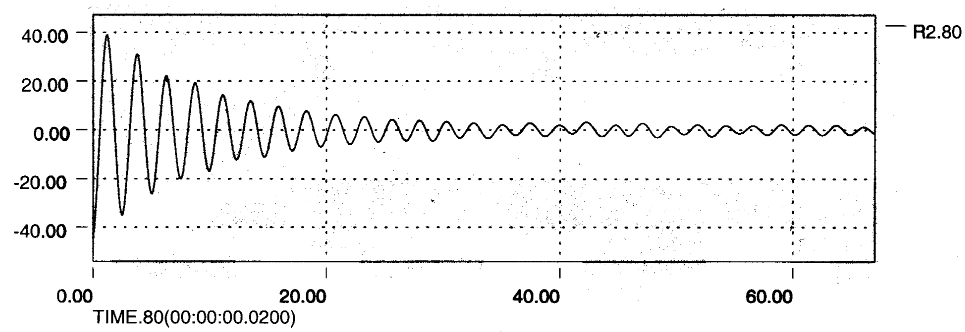

File=20_60_p.xp; Signal Suffix=.60; Date=
File=20_70_p.xp; Signal Suffix=.70; Date=

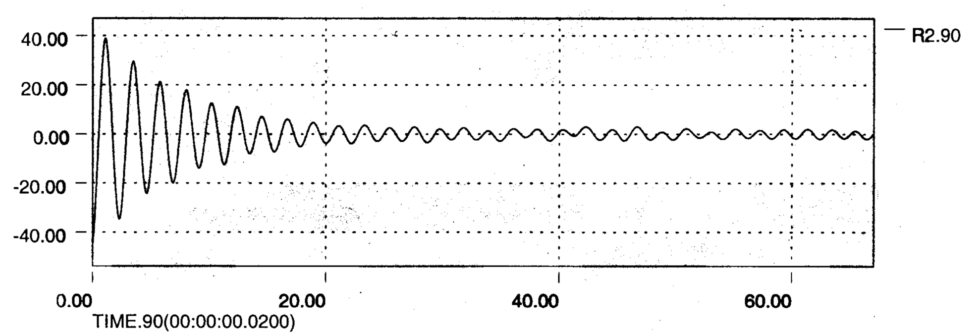
Thu Sep 13 16:49:41 2001

Passive Stabilizer
4X6



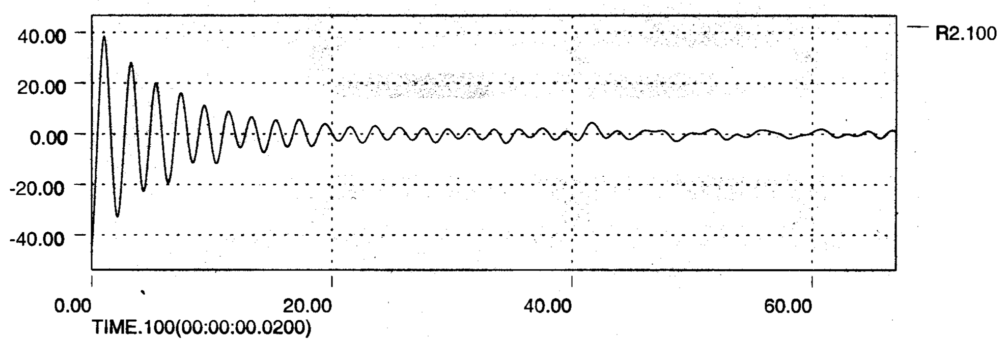



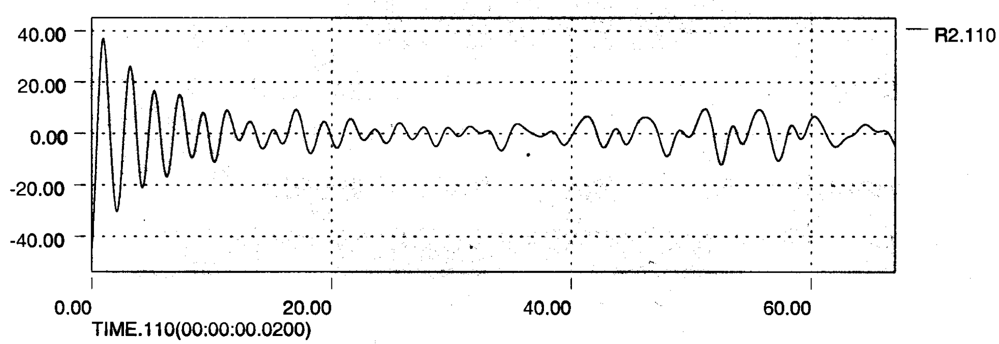



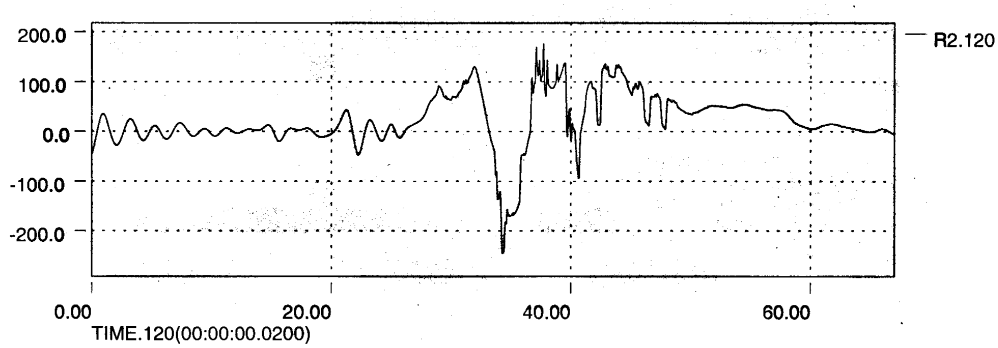

File=24_60_p.xp; Signal Suffix=[none]; Date=
File=24_60_p.xp; Signal Suffix=.60; Date=

Thu Sep 13 16:50:41 2001

Passive Stabilizer
4X6



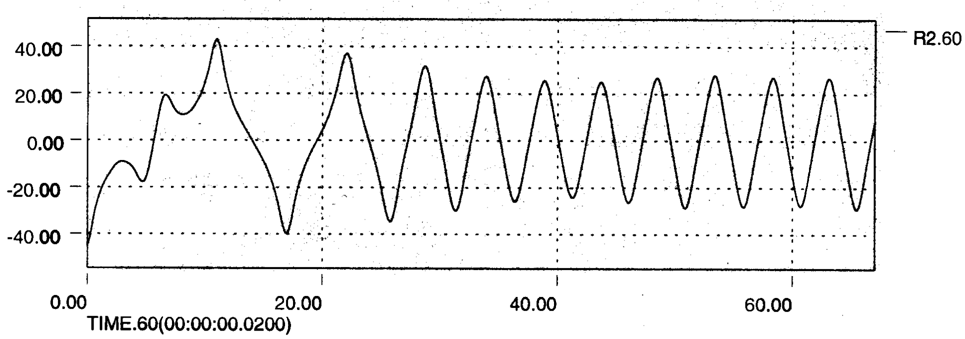



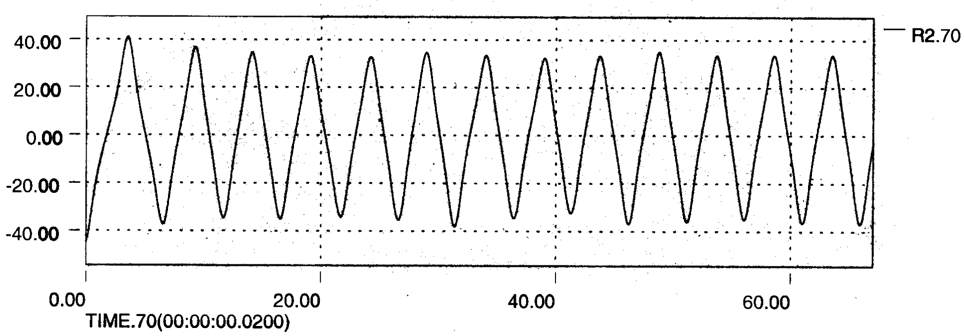



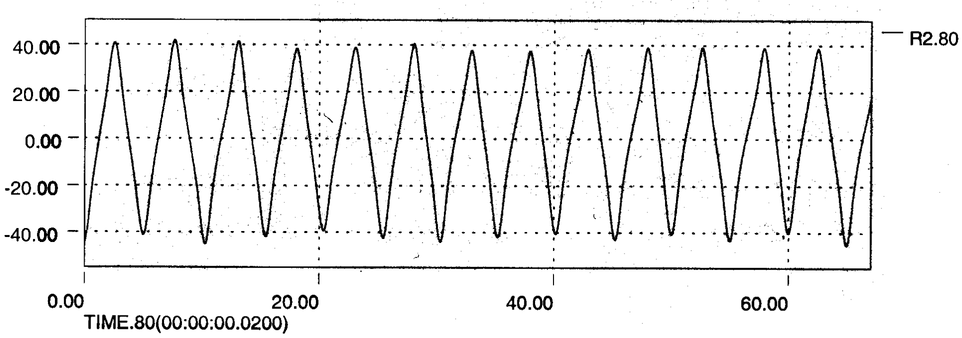

File=24_60_p.xp; Signal Suffix=[none]; Date=
File=24_60_p.xp; Signal Suffix=.60; Date=

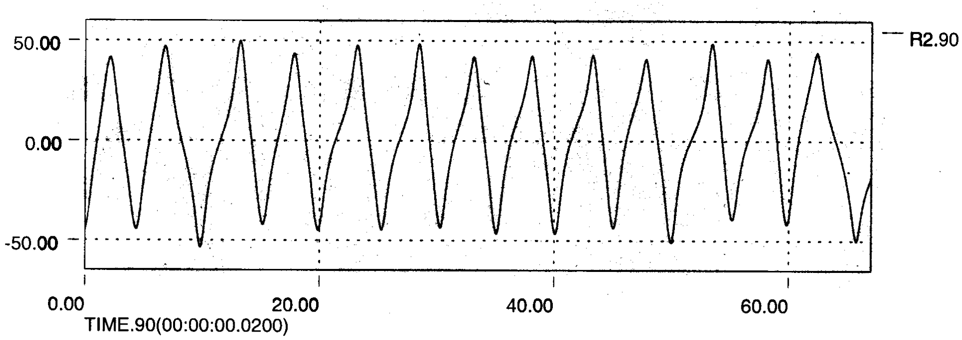
Active Stabilizer
1X3

Thu Sep 13 19:51:59 2001



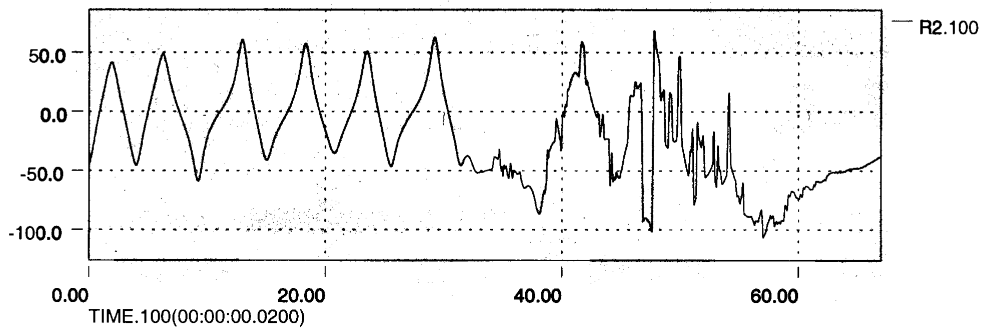



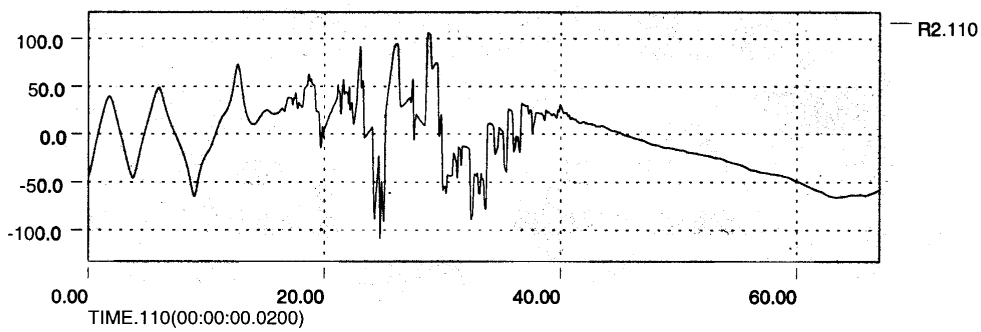



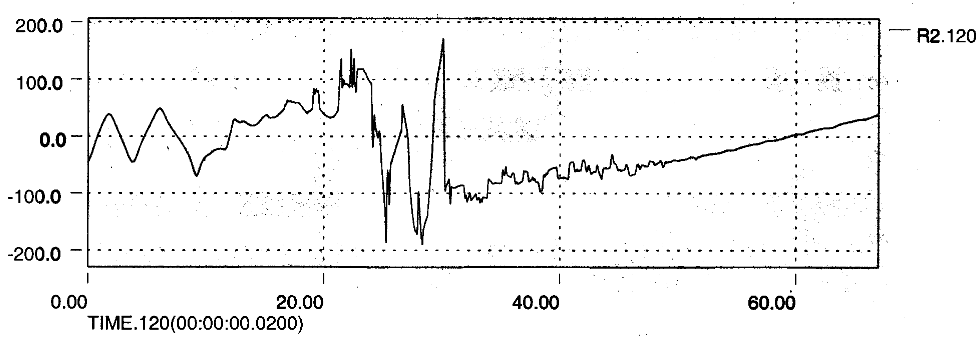

File=3_60_a.xp; Signal Suffix=.60; Date=
File=3_70_a.xp; Signal Suffix=.70; Date=

Active Stabilizer
1X3

Thu Sep 13 19:53:00 2001



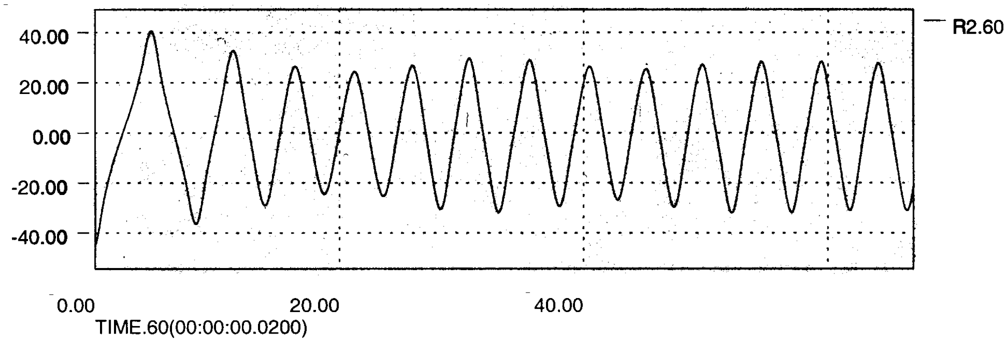



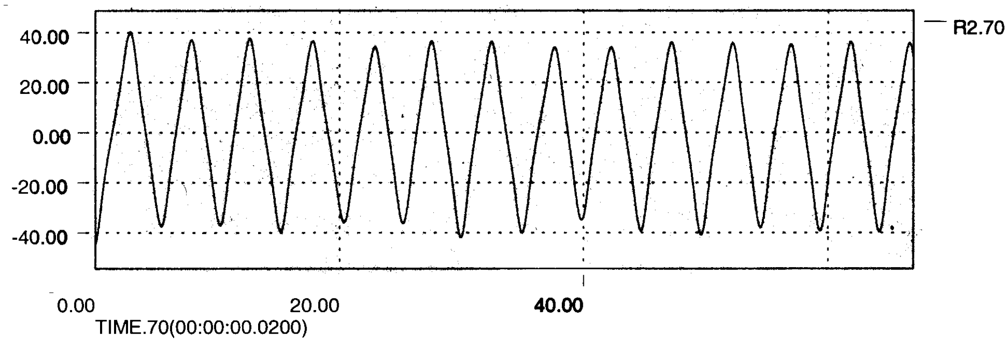



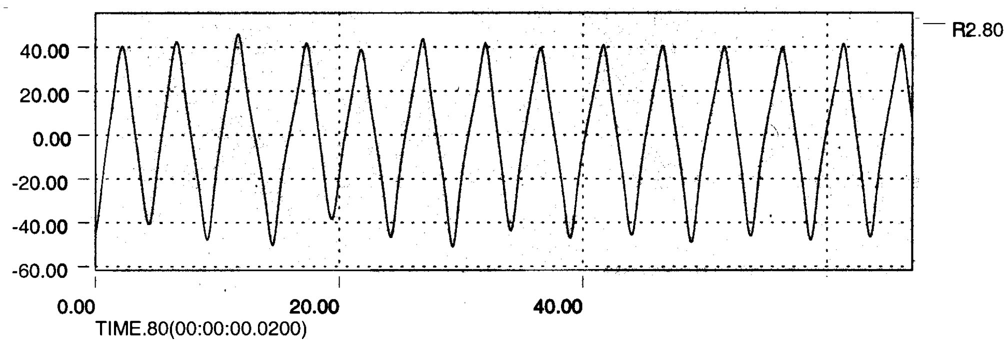

File=3_60_a.xp; Signal Suffix=.60; Date=
File=3_70_a.xp; Signal Suffix=.70; Date=

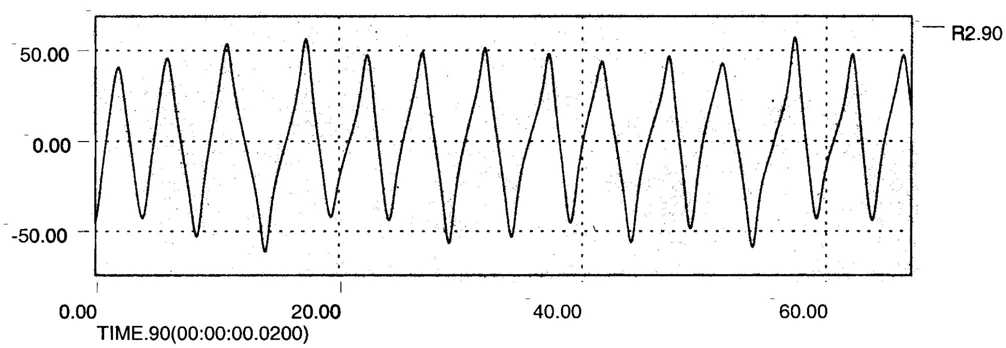
Thu Sep 13 20:02:20 2001

Active Stabilizer
2X3



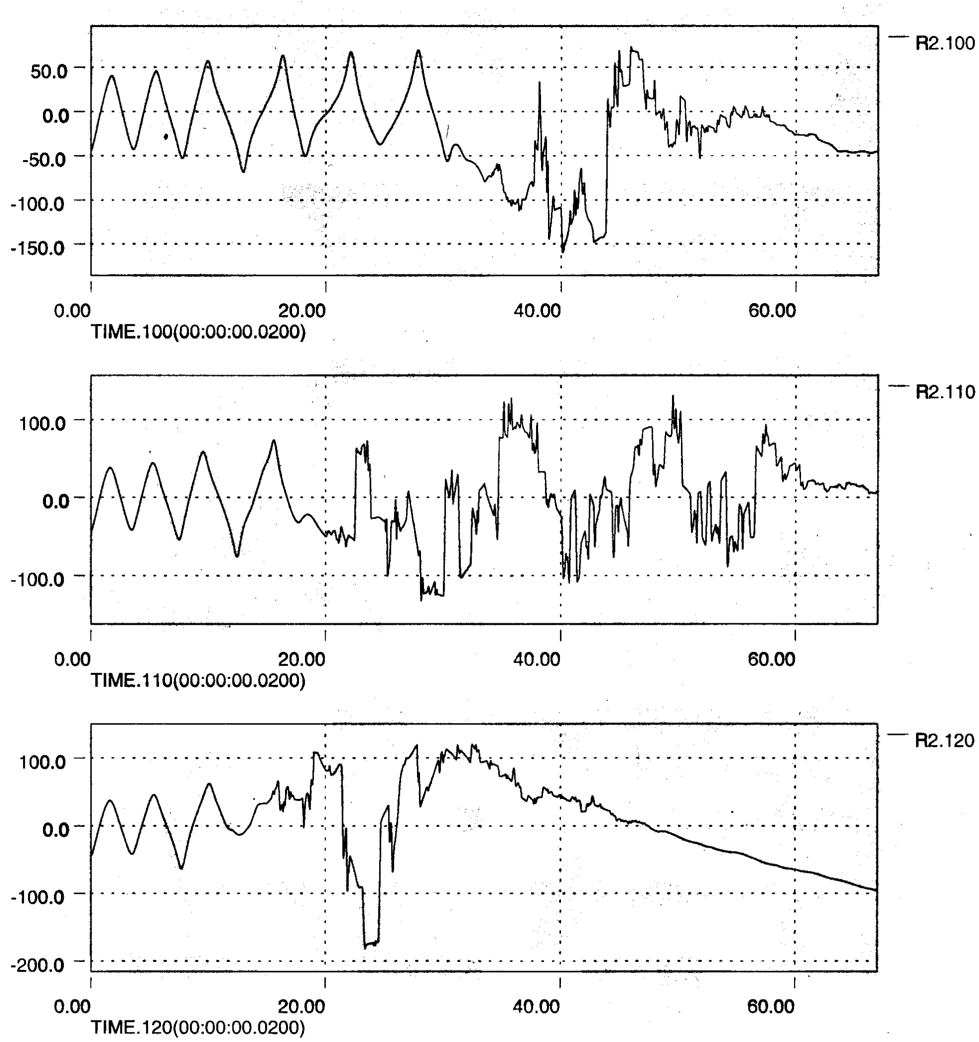




File=6_60_a.xp; Signal Suffix=[none]; Date=
File=6_60_a.xp; Signal Suffix=.60; Date=

Thu Sep 13 20:03:16 2001

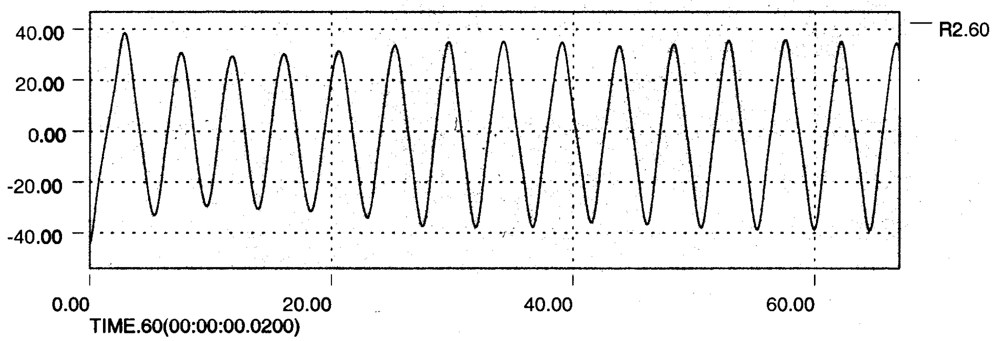
Active Stabilizer
2X3

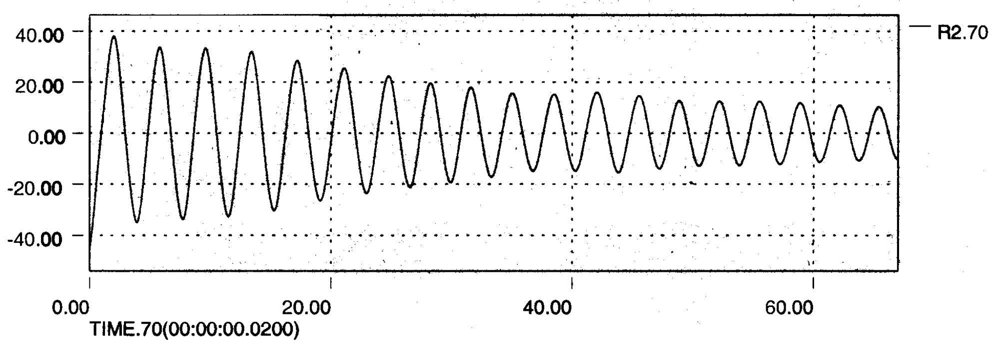


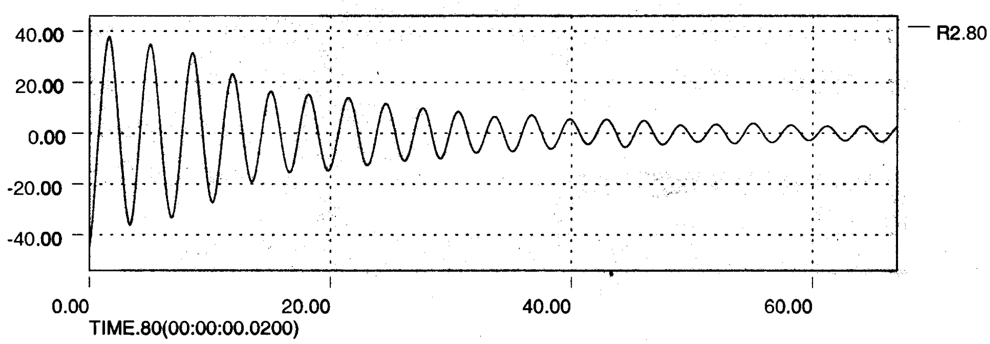

File=6_60_a.xp; Signal Suffix=[none]; Date=
File=6_60_a.xp; Signal Suffix=.60; Date=

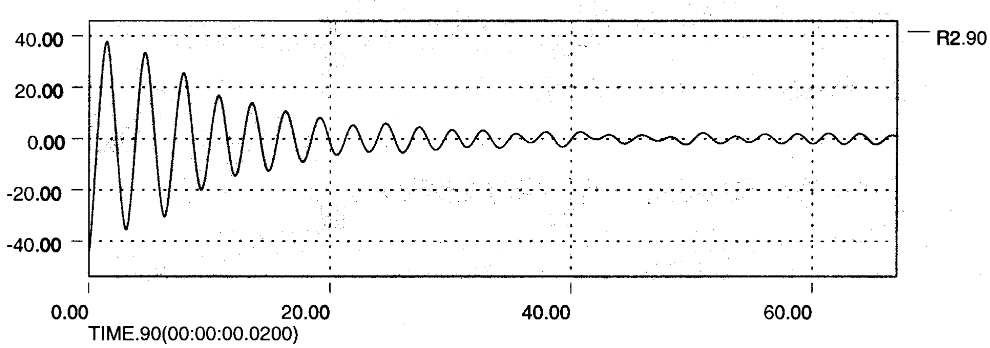
Thu Sep 13 20:11:04 2001

Active Stabilizer
2X4



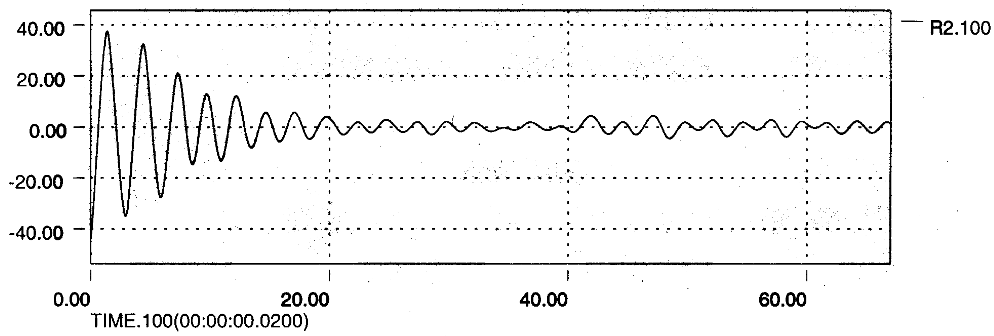



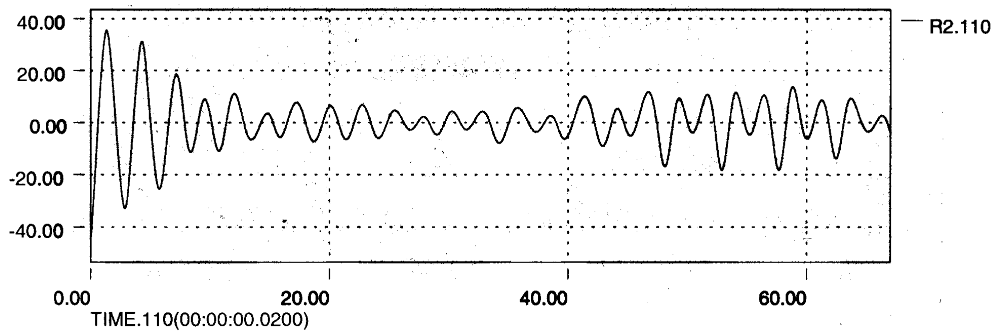



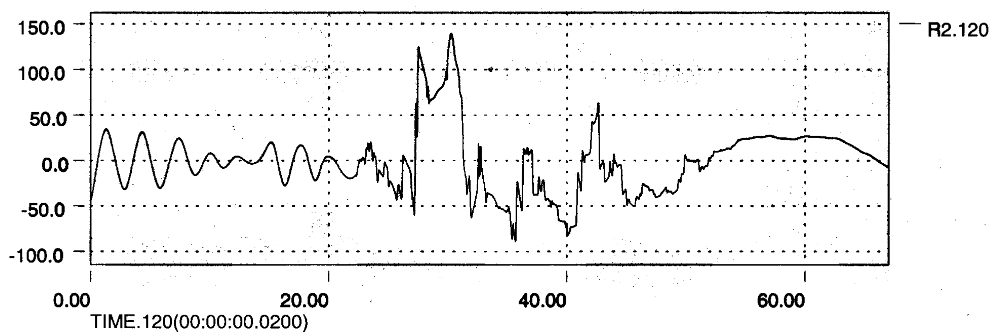

File=8_60_a.xp; Signal Suffix=.60; Date=
File=8_70_a.xp; Signal Suffix=.70; Date=

Thu Sep 13 20:12:06 2001

Active Stabilizer
2X4

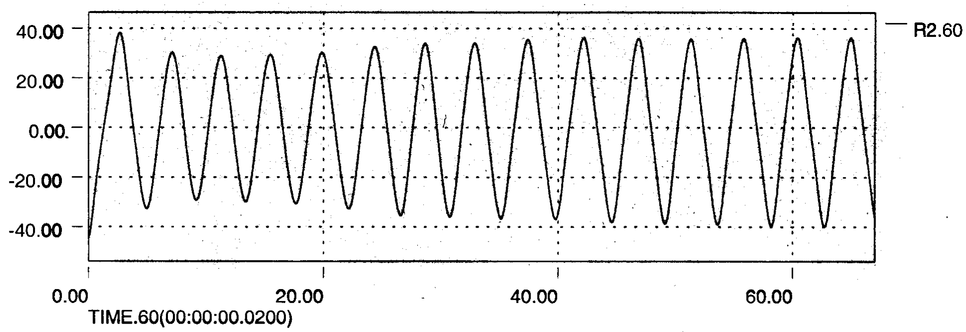


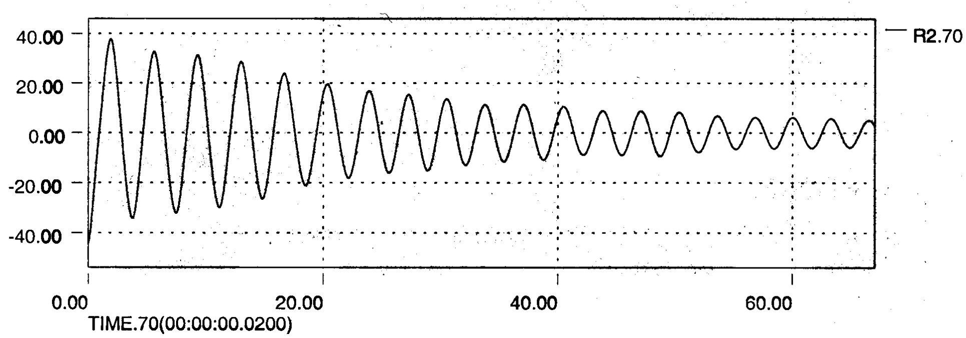



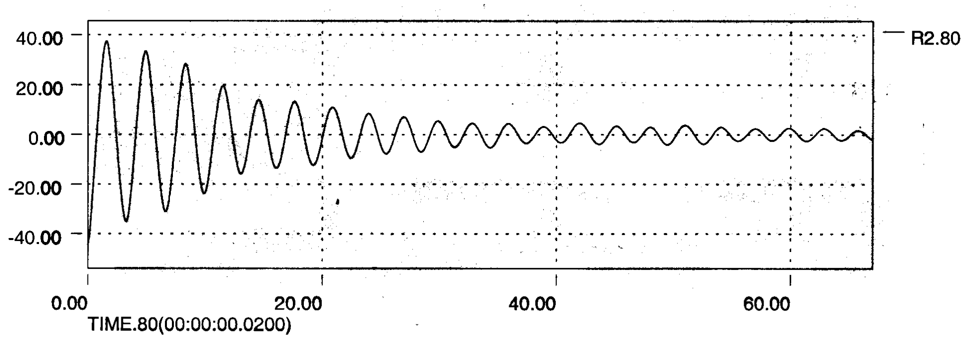



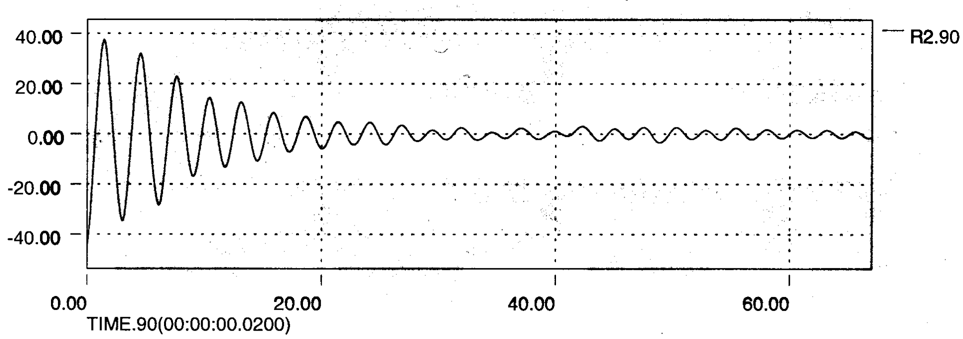

File=8_60_a.xp; Signal Suffix=.60; Date=
File=8_70_a.xp; Signal Suffix=.70; Date=

Active Stabilizer
3X4



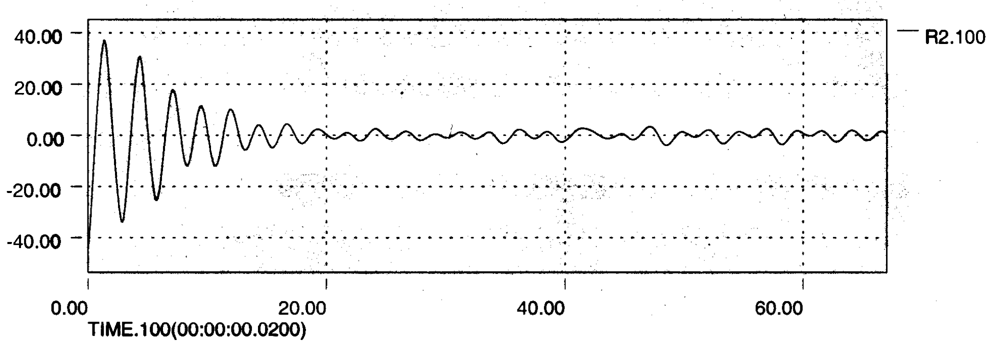



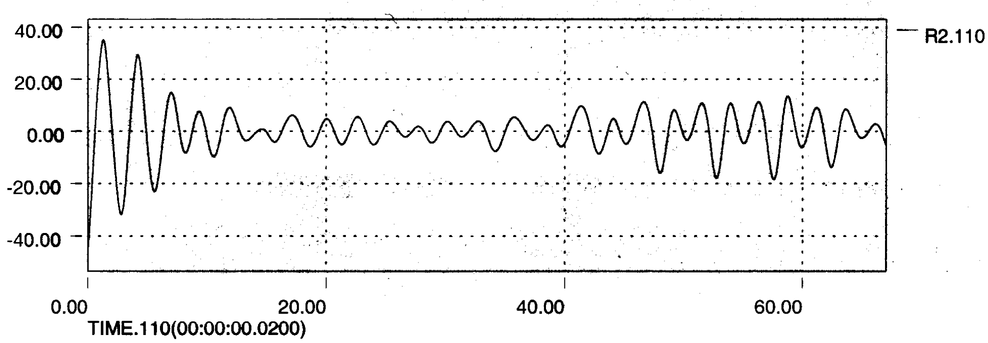



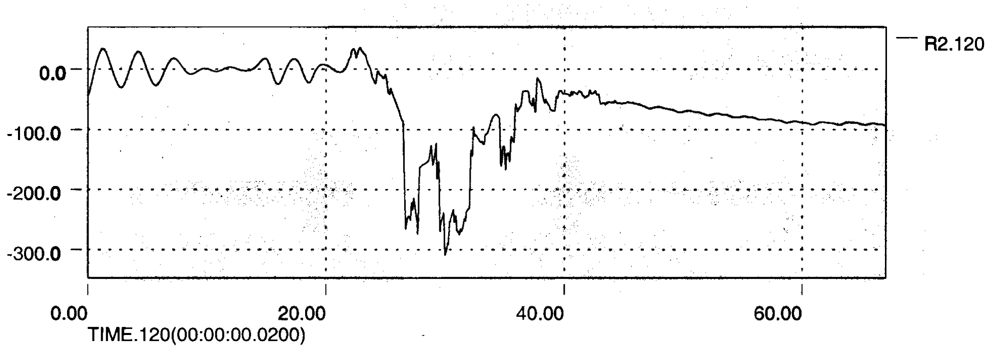

File=12_60_a.xp; Signal Suffix=.60; Date=
File=12_70_a.xp; Signal Suffix=.70; Date=

Thu Sep 13 20:23:38 2001

Active Stabilizer
3X4



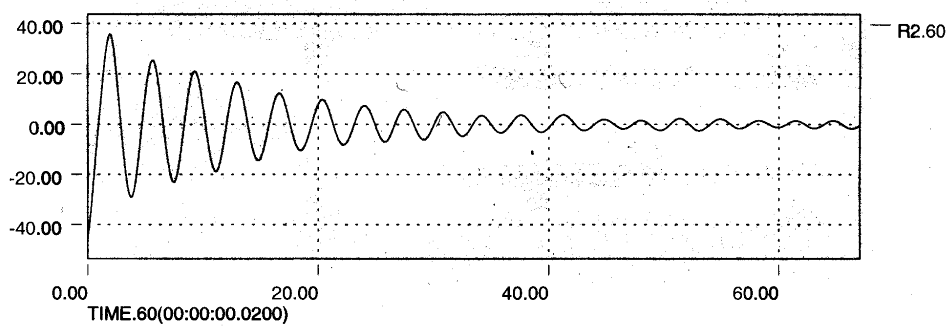



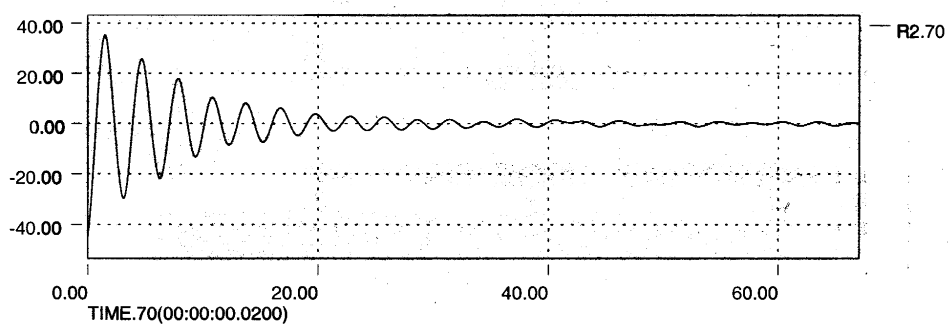



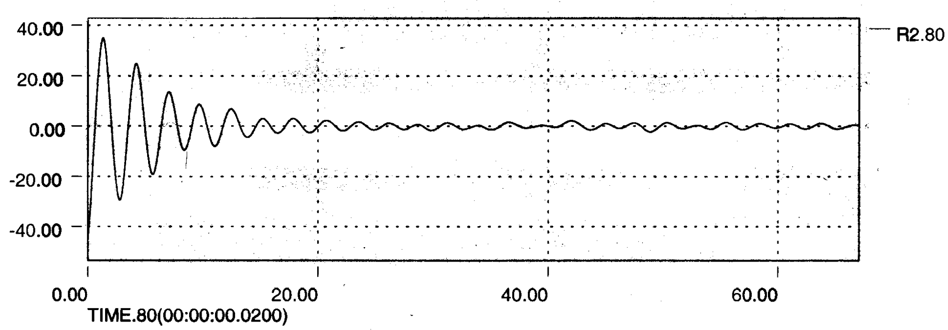

File=12_60_a.xp; Signal Suffix=.60; Date=
File=12_70_a.xp; Signal Suffix=.70; Date=

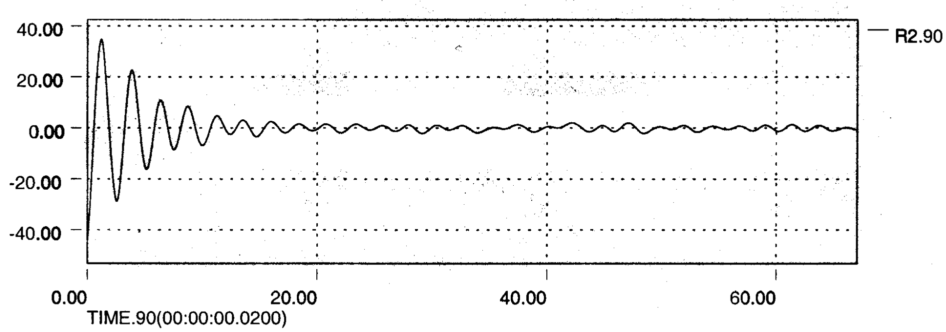
Thu Sep 13 20:30:43 2001

Active Stabilizer
3X5



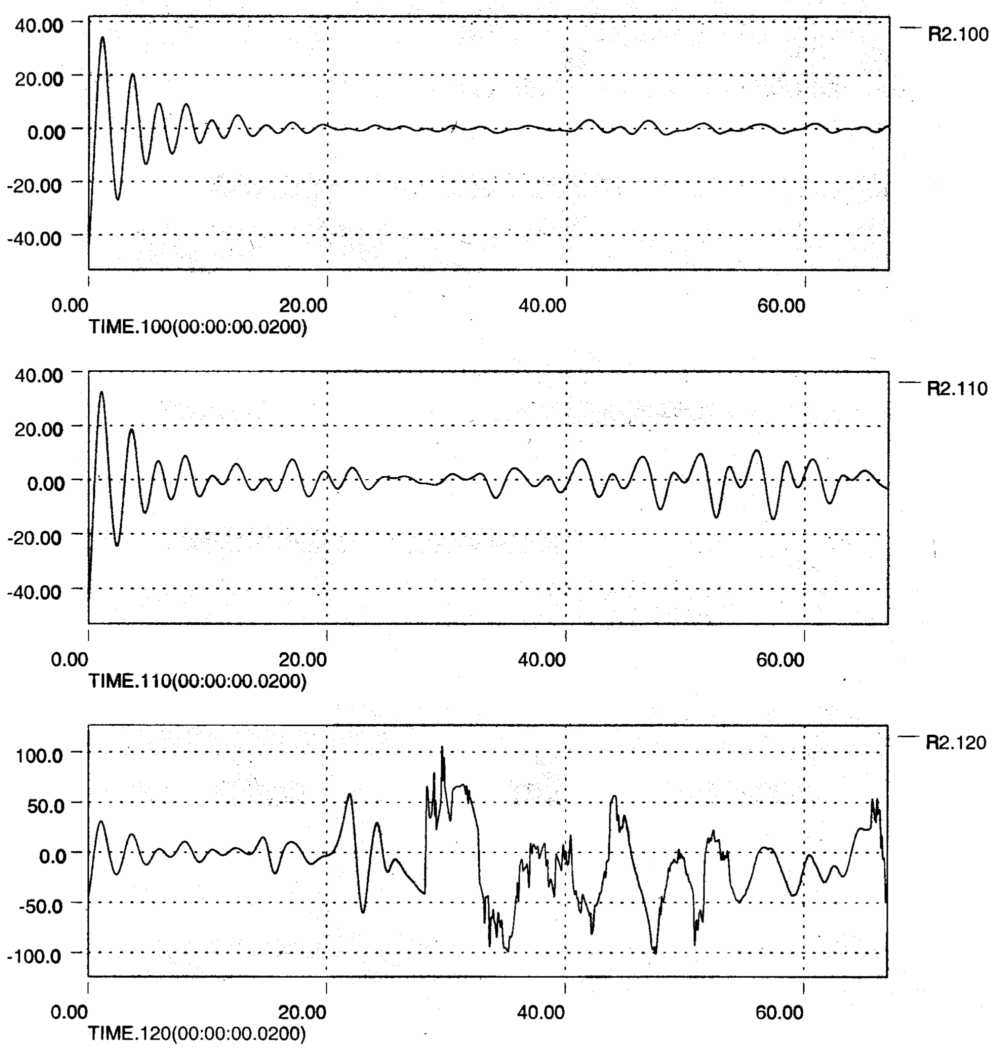




File=15_60_a.xp; Signal Suffix=.60; Date=
File=15_70_a.xp; Signal Suffix=.70; Date=

Thu Sep 13 20:31:54 2001

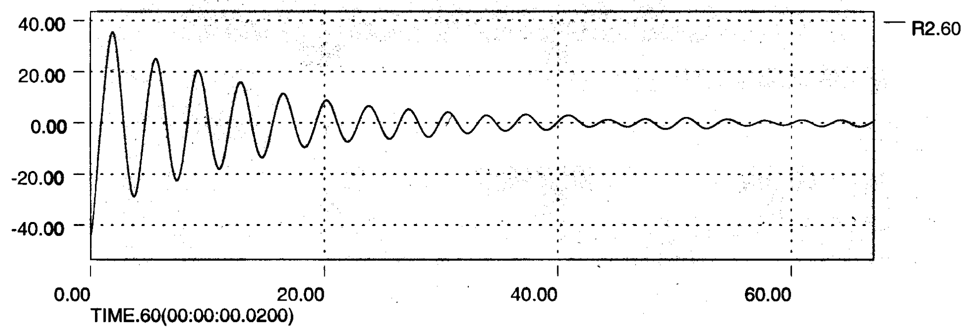
Active Stabilizer
3X5

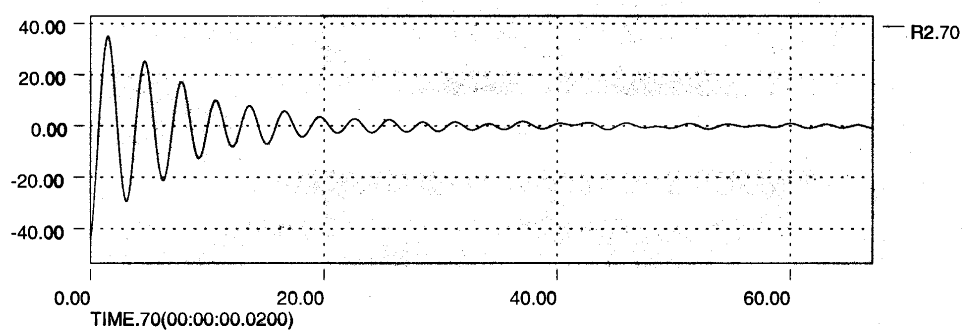


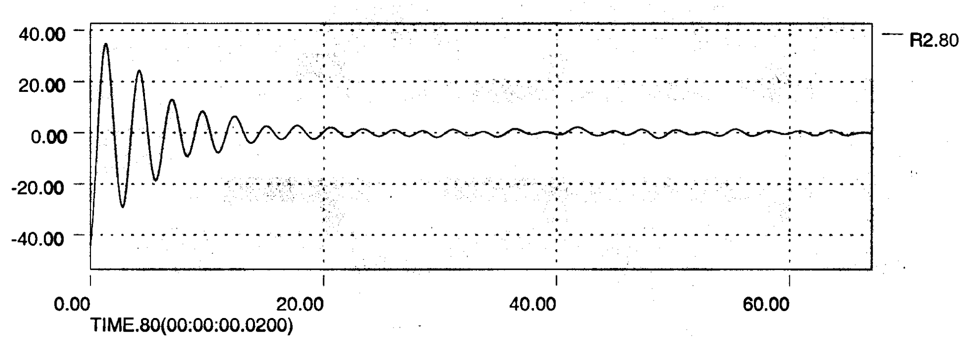

File=15_60_a.xp; Signal Suffix=.60; Date=
File=15_70_a.xp; Signal Suffix=.70; Date=

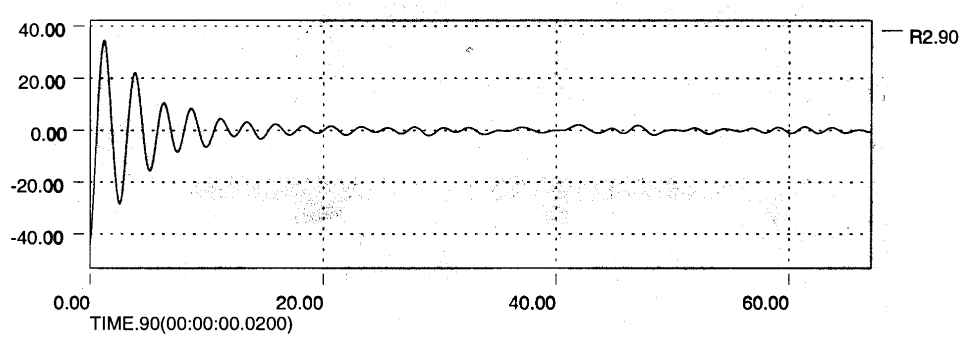
Thu Sep 13 20:42:04 2001

Active Stabilizer
4X5



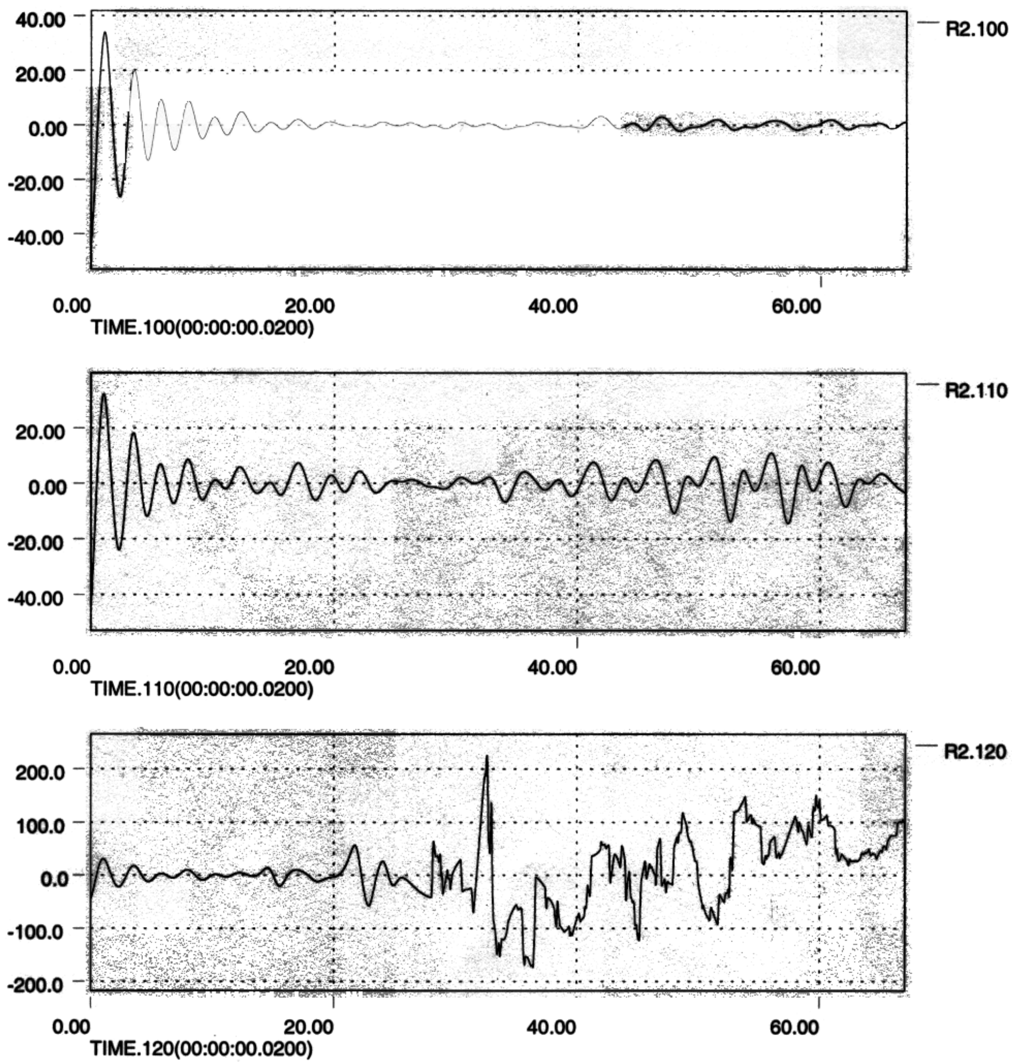




File=20_60_a.xp; Signal Suffix=.60; Date=
File=20_70_a.xp; Signal Suffix=.70; Date=

Thu Sep 13 20:43:17 2001

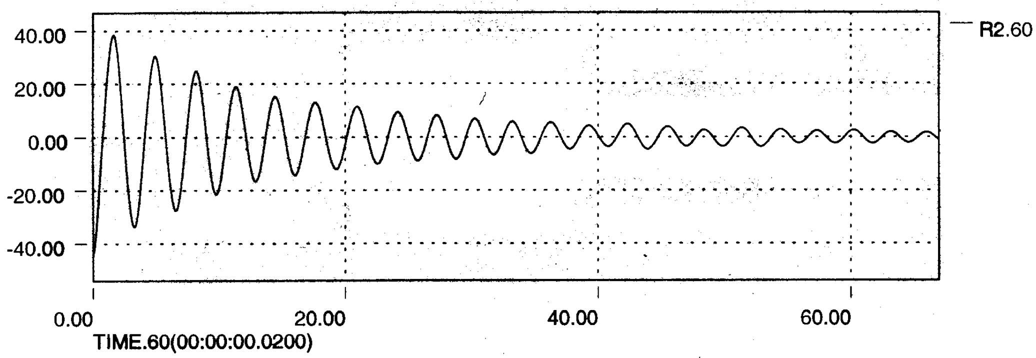
Active Stabilizer
4X5

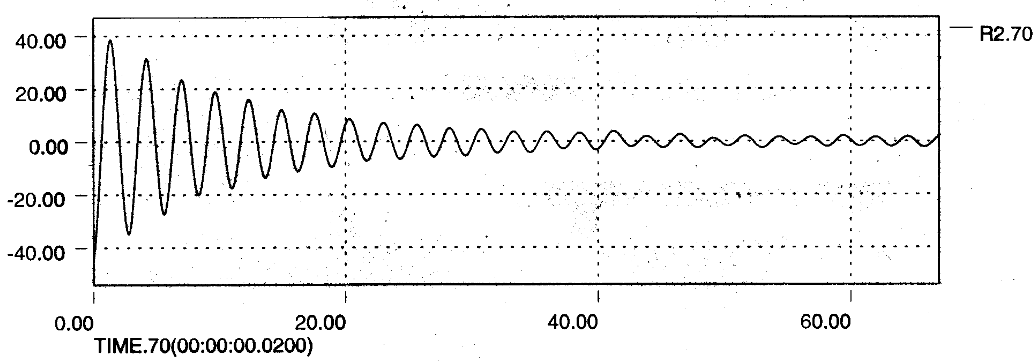


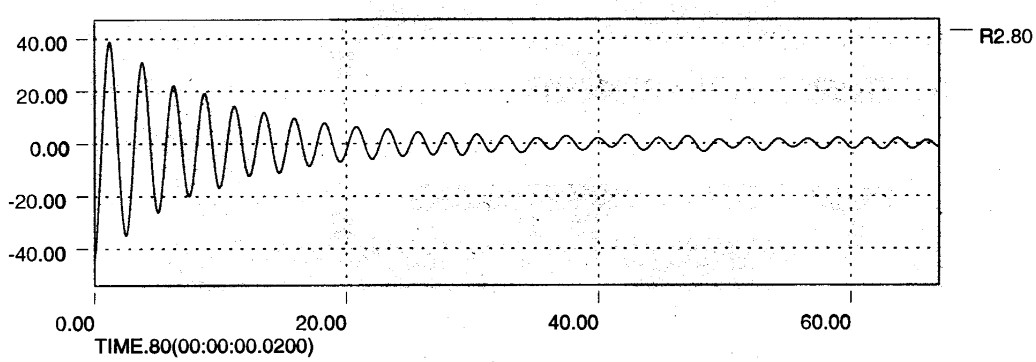

File=20_60_a.xp; Signal Suffix=.60; Date=
File=20_70_a.xp; Signal Suffix=.70; Date=

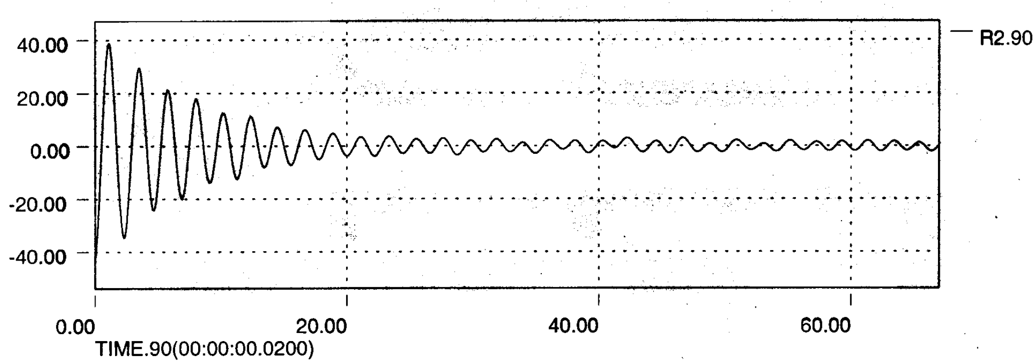
Thu Sep 13 20:49:49 :

Active Stabilizer
4X6



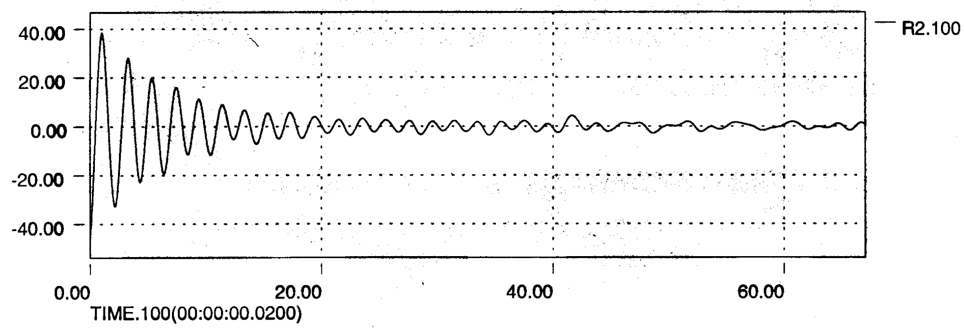



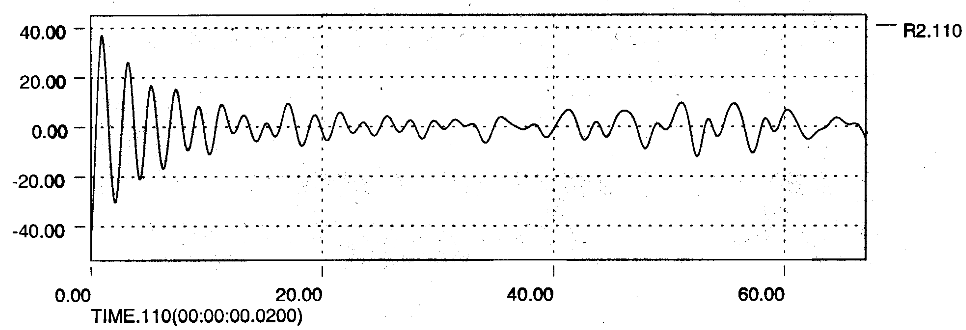



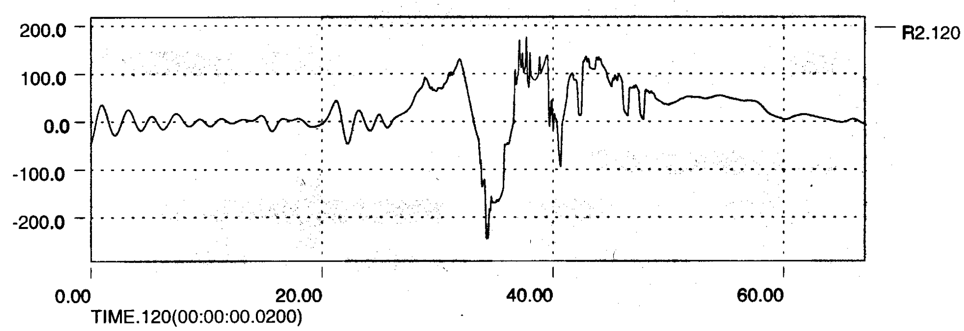

File=24_60_a.xp; Signal Suffix=[none]; Date=
File=24_60_a.xp; Signal Suffix=.60; Date=

Thu Sep 13 20:50:55 2001

Active Stabilizer
4X6




File=24_60_a.xp; Signal Suffix=[none]; Date=
File=24_60_a.xp; Signal Suffix=.60; Date=

## INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Fort Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Marine Corps Representative
   Naval Postgraduate School
   Monterey, California

4. Director, Training and Education, MCCDC, Code C46
   Quantico, Virginia

5. Director, Marine Corps Research Center, MCCDC, Code C40RC
   Quantico, Virginia

6. Marine Corps Tactical Systems Support Activity (Attn: Operations Officer)
   Camp Pendleton, California

7. Chairman Department of Aeronautics and Astronautics
   Naval Postgraduate School
   Monterey, California

8. Department of Aeronautics and Astronautics
   Naval Postgraduate School
   Monterey, California

9. NASA Ames Research Center, Army/NASA Rotorcraft Division